lab6

June 26, 2024

[35]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
from heamy.estimator import Regressor
#     import getargspec   lib/heamy/estimator.py -     estimator
from heamy.pipeline import ModelsPipeline
from heamy.dataset import Dataset
from sklearn.metrics import mean_absolute_error
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
sns.set(style="ticks")
from sklearn.neural_network import MLPRegressor
```

[3]:
```
data = pd.read_csv('Mobile Price Prediction.csv', sep=",")
```

[4]:
```
data.shape
```

[4]:
```
(292, 6)
```

[5]:
```
data.head
```

[5]:
```
<bound method NDFrame.head of      Screen Size  RAM  Storage  Battery Capacity
Camera Quality  Price
0            6.2    4       64               4000             48     500
1            6.1    3      128               3110             12     800
2            6.5    6      256               4500             64     600
3            6.7    8      128               4300             48     700
4            6.4    6      128               4200             40     550
..           …  …      …               …             …    …
287          6.0    6       64               3700             12     750
288          6.3    4      128               4025             32     450
289          6.6    8      256               5000             48     700
290          6.5    6      128               4500             64     550
```

```
291            6.1    4      64               3000                16     400

[292 rows x 6 columns]>
```

```
[6]: data.dtypes
```

```
[6]: Screen Size         float64
     RAM                   int64
     Storage               int64
     Battery Capacity      int64
     Camera Quality        int64
     Price                 int64
     dtype: object
```

1 2.

```
[7]: data.isnull().sum()
```

```
[7]: Screen Size         0
     RAM                 0
     Storage             0
     Battery Capacity    0
     Camera Quality      0
     Price               0
     dtype: int64
```

2 3.

```
[8]: ins_data = data.drop(columns='Price')
     ins_X, ins_y = ins_data, data[['Price']].values
     ins_X_train, ins_X_test, ins_y_train, ins_y_test = train_test_split(
         ins_X, ins_y, test_size=0.2, random_state=1)
```

3 4.1

```
[9]: #
     def val_mae(model):
         model.fit(ins_X_train, ins_y_train)
         y_pred = model.predict(ins_X_test)
         result = mean_absolute_error(ins_y_test, y_pred)
         print(model)
         print('MAE={}'.format(result))
```

[38]:

```
#
for model in [
    LinearRegression(),
    DecisionTreeRegressor(),
    RandomForestRegressor(n_estimators=50)
]:
    val_mae(model)
    print()
```

```
LinearRegression()
MAE=91.77213251714615

DecisionTreeRegressor()
MAE=5.932203389830509

RandomForestRegressor(n_estimators=50)
MAE=10.932203389830509


c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

[36]:

```
#             heamy
#

dataset = Dataset(ins_X_train, ins_y_train, ins_X_test)

#
model_tree = Regressor(dataset=dataset, estimator=DecisionTreeRegressor,␣
  ↪name='tree')
model_lr = Regressor(dataset=dataset, estimator=LinearRegression,name='lr')
model_rf = Regressor(dataset=dataset, estimator=RandomForestRegressor,␣
  ↪parameters={'n_estimators': 50},name='rf')
```

[12]:

```
#         -       :
#        :

pipeline = ModelsPipeline(model_tree, model_lr)
stack_ds = pipeline.stack(k=10, seed=1)
#
stacker = Regressor(dataset=stack_ds, estimator=LinearRegression)
results = stacker.validate(k=10,scorer=mean_absolute_error)
```

```
Metric: mean_absolute_error
Folds accuracy: [10.099482322212891, 11.079378673785781, 19.34684684227087,
30.209752520471028, 9.873620825195745, 21.754293964323068, 12.3393354483684,
24.53804227710834, 8.258690680427428, 22.413285340984103]
Mean accuracy: 16.991272889514768
Standard Deviation: 7.213334906396753
Variance: 52.03220047184185
```

[37]:

```
#          -       :    ,
#         :
pipeline = ModelsPipeline(model_tree, model_lr, model_rf)
stack_ds3 = pipeline.stack(k=10, seed=1)
#
stacker = Regressor(dataset=stack_ds3, estimator=LinearRegression)
results = stacker.validate(k=10,scorer=mean_absolute_error)
```

```
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
```

```
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\base.py:1474: DataConversionWarning: A column-vector y was
passed when a 1d array was expected. Please change the shape of y to
(n_samples,), for example using ravel().
  return fit_method(estimator, *args, **kwargs)
```

```
Metric: mean_absolute_error
Folds accuracy: [13.589403717629898, 11.54047104518372, 11.851948191333255,
30.949803198842545, 8.597168913783637, 14.7265943482436, 12.32657560629455,
25.638230662453452, 11.743345265394503, 15.05109971590865]
Mean accuracy: 15.601464066506784
Standard Deviation: 6.683154509112947
Variance: 44.66455419267672
```

[39]:

```
#          -       :
#         :

pipeline = ModelsPipeline(model_tree, model_lr)
stack_ds = pipeline.stack(k=10, seed=1)
#
stacker = Regressor(dataset=stack_ds, estimator=DecisionTreeRegressor)
results = stacker.validate(k=10,scorer=mean_absolute_error)
```

```
Metric: mean_absolute_error
Folds accuracy: [16.666666666666668, 22.916666666666668, 12.5,
32.608695652173914, 4.3478260869565215, 17.391304347826086, 8.695652173913043,
19.565217391304348, 17.391304347826086, 17.391304347826086]
Mean accuracy: 16.947463768115945
Standard Deviation: 7.314719612167022
Variance: 53.50512300462087
```

[41]:

```
#           -       :    ,
#          :

pipeline = ModelsPipeline(model_tree, model_rf, model_lr)
stack_ds = pipeline.stack(k=10, seed=1)
#
stacker = Regressor(dataset=stack_ds, estimator=DecisionTreeRegressor)
results = stacker.validate(k=10,scorer=mean_absolute_error)
```

```
Metric: mean_absolute_error
Folds accuracy: [8.333333333333334, 6.25, 0.0, 28.26086956521739,
4.3478260869565215, 13.043478260869565, 15.217391304347826, 21.73913043478261,
17.391304347826086, 10.869565217391305]
Mean accuracy: 12.545289855072465
Standard Deviation: 8.044503337540489
Variance: 64.71403394770006
```

[42]:

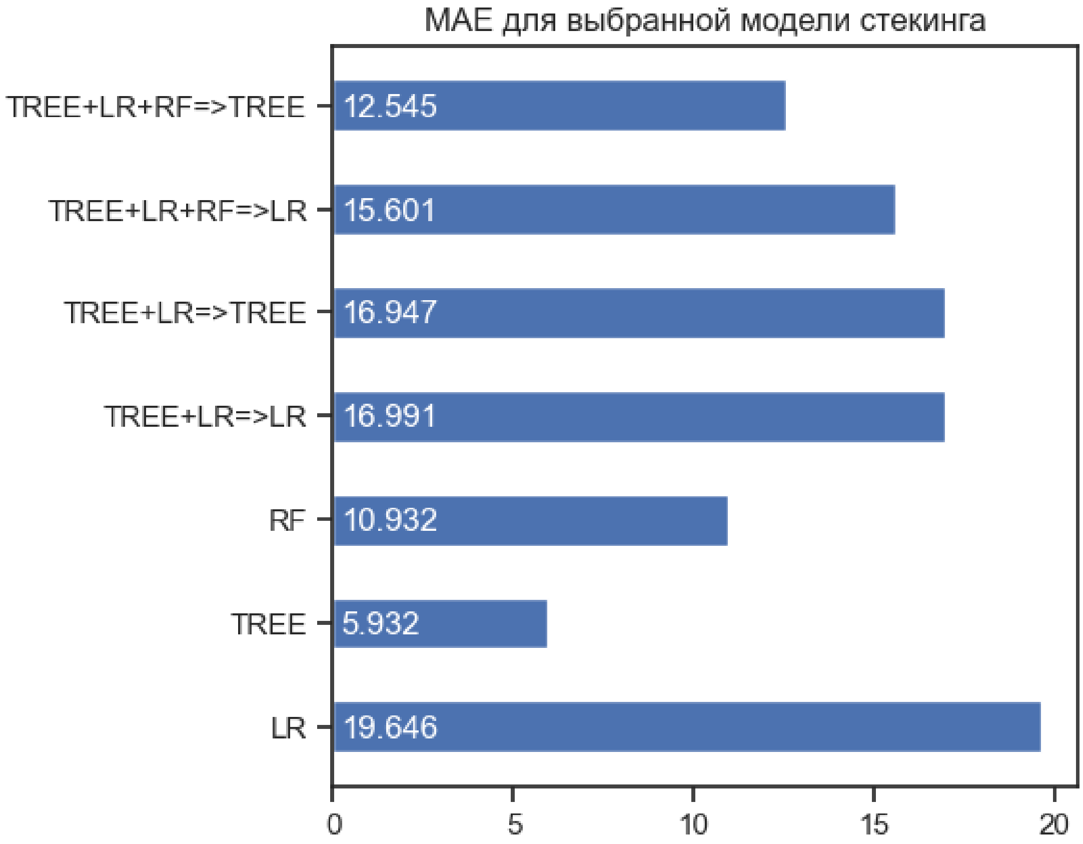
```
#
array_labels = ['LR', 'TREE', 'RF', 'TREE+LR=>LR', 'TREE+LR=>TREE',␣
 ↪'TREE+LR+RF=>LR', 'TREE+LR+RF=>TREE']
array_mae = [19.645947869043603,
             5.932203389830509,
             10.932203389830509,
             16.991272889514768,
             16.947463768115945,
             15.601464066506784,
             12.545289855072465]
```

[43]:

```
def vis_models_quality(array_metric, array_labels, str_header, figsize=(5, 5)):
    fig, ax1 = plt.subplots(figsize=figsize)
    pos = np.arange(len(array_metric))
    rects = ax1.barh(pos, array_metric,
                     align='center',
                     height=0.5,
                     tick_label=array_labels)
    ax1.set_title(str_header)
    for a,b in zip(pos, array_metric):
        plt.text(0.2, a-0.1, str(round(b,3)), color='white')
    plt.show()
```

[44]:

```
#
vis_models_quality(array_mae, array_labels, 'MAE                  ')
```



4 4.2

[16]:

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

#
ins_X_train_scaled = scaler.fit_transform(ins_X_train)
ins_X_test_scaled = scaler.fit_transform(ins_X_test)
```

[17]:

```
from time import time

i = 0
df = pd.DataFrame(columns =␣
 ↪['alpha','max_iter','train_mae','test_mae','train_time'])
for a in [0.00001,0.0001,0.001,0.01, 0.1, 1, 10]:
    for mi in [10, 100, 1000, 10000]:
        st = time()
        mlp = MLPRegressor(alpha=a, max_iter=mi, random_state=1)
        mlp.fit(ins_X_train_scaled, ins_y_train)
        end = time() - st

        mae_tr = mean_absolute_error(ins_y_train, mlp.
 ↪predict(ins_X_train_scaled))
        mae = mean_absolute_error(ins_y_test, mlp.predict(ins_X_test_scaled))
        df.loc[i] = [a,mi,mae_tr,mae,end]
        i=i+1
```

```
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
```

```
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
```

```
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

```
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
```

```
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
```

```
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (100) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (1000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[18]:

```
df.sort_values(by='test_mae').head(3)
#                 19: alpha = 0.1, max_iter = 10000
```

[18]:

```
       alpha  max_iter  train_mae   test_mae  train_time
19  0.10000   10000.0  44.598683  66.395082    5.492069
3   0.00001   10000.0  48.785653  68.379520    6.076551
7   0.00010   10000.0  48.781782  68.387106    6.130107
```

[19]:

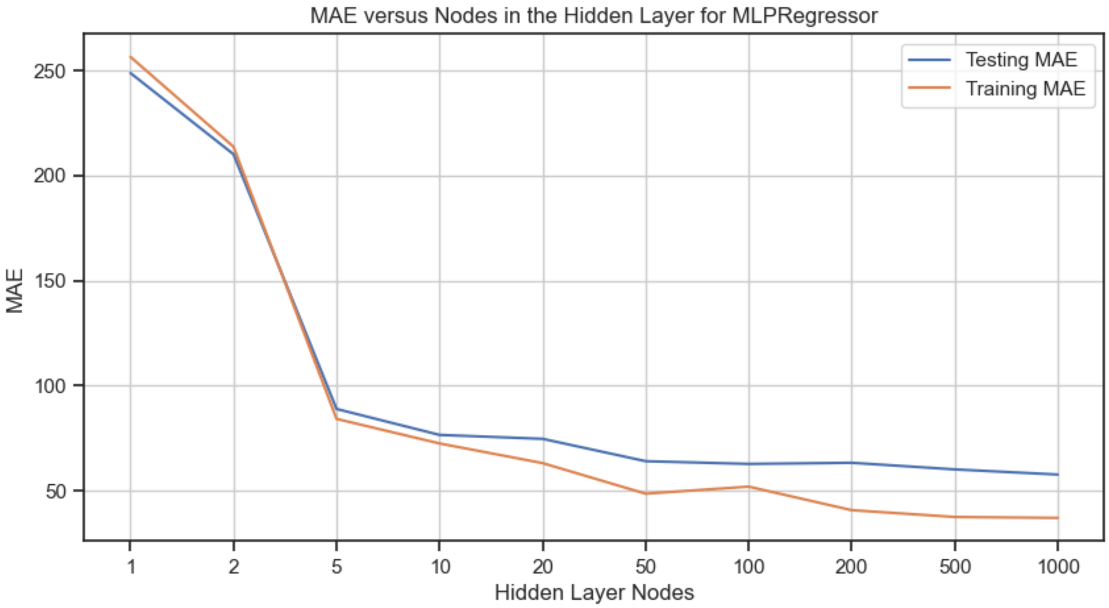
```
mae = []
mae_tr = []
timelog = []
for l in [1,2,5,10,20,50,100,200,500,1000]:
    t = time()
    mlp = MLPRegressor(alpha=0.1, max_iter=10000, hidden_layer_sizes=(l,))
    mlp.fit(ins_X_train_scaled, ins_y_train)
    endt = time() - t

    m_tr = mean_absolute_error(ins_y_train, mlp.predict(ins_X_train_scaled))
    m = mean_absolute_error(ins_y_test, mlp.predict(ins_X_test_scaled))

    mae_tr.append(m_tr)
    mae.append(m)
    timelog.append(endt)
```

```
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
```

```
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:691:
ConvergenceWarning: Stochastic Optimizer: Maximum iterations (10000) reached and
the optimization hasn't converged yet.
  warnings.warn(
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
```

```
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[20]:
```
l = [1,2,5,10,20,50,100,200,500,1000]
N = len(l)
l2 = np.arange(N)
plt.subplots(figsize=(10, 5))
plt.plot(l2, mae, label="Testing MAE")
plt.plot(l2, mae_tr, label="Training MAE")
plt.xticks(l2,l)
plt.grid(True)
plt.xlabel("Hidden Layer Nodes")
plt.ylabel("MAE")
plt.legend()
plt.title('MAE versus Nodes in the Hidden Layer for MLPRegressor', fontsize=12)
plt.show()
```



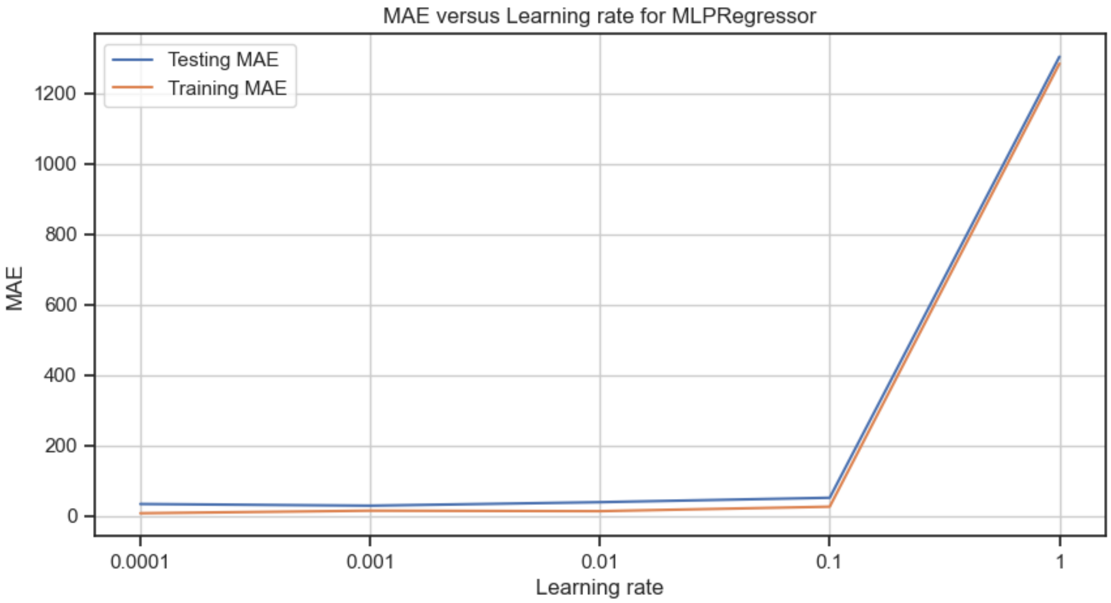
[21]:

```
mae = []
mae_tr = []
timelog = []
for l in [0.0001, 0.001, 0.01, 0.1, 1]:
    t = time()
    mlp = MLPRegressor(alpha=0.1, max_iter=10000,␣
  ↪hidden_layer_sizes=(1000,500), learning_rate_init=l)
    mlp.fit(ins_X_train_scaled, ins_y_train)
    endt = time() - t

    m_tr = mean_absolute_error(ins_y_train, mlp.predict(ins_X_train_scaled))
    m = mean_absolute_error(ins_y_test, mlp.predict(ins_X_test_scaled))

    mae_tr.append(m_tr)
    mae.append(m)
    timelog.append(endt)
```

```
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

[22]:

```
l = [0.0001, 0.001, 0.01, 0.1, 1]
N = len(l)
l2 = np.arange(N)
plt.subplots(figsize=(10, 5))
plt.plot(l2, mae, label="Testing MAE")
plt.plot(l2, mae_tr, label="Training MAE")
plt.xticks(l2,l)
plt.grid(True)
plt.xlabel("Learning rate")
plt.ylabel("MAE")
plt.legend()
plt.title('MAE versus Learning rate for MLPRegressor', fontsize=12)
plt.show()
```



[23]:

```
mlp = MLPRegressor(hidden_layer_sizes=(1000, 500), max_iter=10000, alpha=0.1,
                   solver='adam', verbose=10, random_state=1,␣
 ↪learning_rate_init=0.001)
#     sgd        adam -        loss

mlp.fit(ins_X_train_scaled, ins_y_train)
```

```
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\neural_network\_multilayer_perceptron.py:1624:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
  y = column_or_1d(y, warn=True)
```

```
Iteration 1, loss = 204434.15083671
Iteration 2, loss = 203587.73005234
Iteration 3, loss = 202708.55171288
Iteration 4, loss = 201702.70549348
Iteration 5, loss = 200480.13033299
Iteration 6, loss = 199031.81753998
Iteration 7, loss = 197322.61110278
Iteration 8, loss = 195336.24980220
Iteration 9, loss = 193028.48457161
Iteration 10, loss = 190364.19311812
Iteration 11, loss = 187301.67810375
Iteration 12, loss = 183798.61161643
Iteration 13, loss = 179845.58572269
Iteration 14, loss = 175359.10704415
Iteration 15, loss = 170371.90285767
Iteration 16, loss = 164805.02148264
Iteration 17, loss = 158659.82927263
Iteration 18, loss = 151923.26699656
Iteration 19, loss = 144550.75259703
Iteration 20, loss = 136631.82318289
Iteration 21, loss = 128201.25653878
Iteration 22, loss = 119220.17775409
Iteration 23, loss = 109763.68897935
Iteration 24, loss = 100148.83179483
Iteration 25, loss = 90198.58328084
Iteration 26, loss = 80359.73616784
Iteration 27, loss = 70627.40572058
Iteration 28, loss = 61278.54471954
Iteration 29, loss = 52663.29713457
Iteration 30, loss = 44784.78271408
Iteration 31, loss = 38277.74206021
Iteration 32, loss = 32904.08174064
Iteration 33, loss = 28979.52345428
Iteration 34, loss = 26432.24364880
Iteration 35, loss = 24983.90978101
Iteration 36, loss = 24498.57049907
Iteration 37, loss = 24652.18080046
Iteration 38, loss = 24588.08223369
Iteration 39, loss = 24634.95002524
Iteration 40, loss = 24382.53014275
Iteration 41, loss = 23745.69930434
Iteration 42, loss = 22876.34894052
Iteration 43, loss = 21962.14558319
Iteration 44, loss = 20963.87191893
Iteration 45, loss = 20155.49296867
Iteration 46, loss = 19417.23302753
Iteration 47, loss = 18813.03485346
Iteration 48, loss = 18236.38712488
Iteration 49, loss = 17788.58394442
Iteration 50, loss = 17351.72035459
Iteration 51, loss = 16935.08650754
Iteration 52, loss = 16532.19828696
Iteration 53, loss = 16120.81440040
Iteration 54, loss = 15710.70425841
Iteration 55, loss = 15299.12940179
Iteration 56, loss = 14904.65815549
Iteration 57, loss = 14533.53199925
Iteration 58, loss = 14170.12994945
Iteration 59, loss = 13830.81473430
Iteration 60, loss = 13496.75471921
Iteration 61, loss = 13182.15654645
Iteration 62, loss = 12875.90284089
Iteration 63, loss = 12574.32992843
Iteration 64, loss = 12289.98093967
Iteration 65, loss = 11998.53939925
Iteration 66, loss = 11728.48678333
Iteration 67, loss = 11498.09755621
Iteration 68, loss = 11249.83125333
Iteration 69, loss = 10989.01668244
Iteration 70, loss = 10739.50643223
Iteration 71, loss = 10493.11744133
Iteration 72, loss = 10278.79260213
Iteration 73, loss = 10065.27995326
Iteration 74, loss = 9858.43708859
Iteration 75, loss = 9656.28401789
Iteration 76, loss = 9460.48844878
Iteration 77, loss = 9276.90929947
Iteration 78, loss = 9105.00853242
Iteration 79, loss = 8935.87037536
Iteration 80, loss = 8781.15919675
Iteration 81, loss = 8625.35905514
Iteration 82, loss = 8461.06749067
Iteration 83, loss = 8305.92271427
Iteration 84, loss = 8163.08871658
Iteration 85, loss = 8024.76722230
Iteration 86, loss = 7881.44435149
Iteration 87, loss = 7758.71473283
Iteration 88, loss = 7639.98101575
Iteration 89, loss = 7515.58741539
Iteration 90, loss = 7392.03048929
Iteration 91, loss = 7274.85751367
Iteration 92, loss = 7158.24396049
Iteration 93, loss = 7046.89925766
Iteration 94, loss = 6942.22136487
Iteration 95, loss = 6842.09502043
Iteration 96, loss = 6740.95422342
Iteration 97, loss = 6641.34957941
Iteration 98, loss = 6547.40828077
Iteration 99, loss = 6457.51293651
Iteration 100, loss = 6368.58162714
Iteration 101, loss = 6282.18909821
Iteration 102, loss = 6207.33259898
Iteration 103, loss = 6126.02658116
Iteration 104, loss = 6059.11605224
Iteration 105, loss = 5983.27798575
Iteration 106, loss = 5918.79110129
Iteration 107, loss = 5851.98515578
Iteration 108, loss = 5784.52842610
Iteration 109, loss = 5720.61923110
Iteration 110, loss = 5661.23827595
Iteration 111, loss = 5604.22296477
Iteration 112, loss = 5547.52859739
Iteration 113, loss = 5491.83307996
Iteration 114, loss = 5437.40687796
Iteration 115, loss = 5385.79202572
Iteration 116, loss = 5329.37496436
Iteration 117, loss = 5279.79650017
Iteration 118, loss = 5225.48801191
Iteration 119, loss = 5179.40390131
Iteration 120, loss = 5133.84203188
Iteration 121, loss = 5094.89973547
Iteration 122, loss = 5047.58829322
Iteration 123, loss = 5009.27010376
Iteration 124, loss = 4959.89874297
Iteration 125, loss = 4918.54932327
Iteration 126, loss = 4876.49650525
Iteration 127, loss = 4839.32762887
Iteration 128, loss = 4809.06511428
Iteration 129, loss = 4772.73919592
Iteration 130, loss = 4734.51597392
Iteration 131, loss = 4692.51731660
Iteration 132, loss = 4654.91124553
Iteration 133, loss = 4619.42905910
Iteration 134, loss = 4582.29836565
Iteration 135, loss = 4547.88718887
Iteration 136, loss = 4511.73171701
Iteration 137, loss = 4478.94397307
Iteration 138, loss = 4452.34907167
Iteration 139, loss = 4425.54899923
Iteration 140, loss = 4407.03480621
Iteration 141, loss = 4384.49240752
Iteration 142, loss = 4356.81928526
Iteration 143, loss = 4318.75681715
Iteration 144, loss = 4282.51527550
Iteration 145, loss = 4255.56618597
Iteration 146, loss = 4227.18616866
Iteration 147, loss = 4207.79558727
Iteration 148, loss = 4182.86909102
Iteration 149, loss = 4156.29647847
Iteration 150, loss = 4126.00432046
Iteration 151, loss = 4111.78402503
Iteration 152, loss = 4083.51570767
Iteration 153, loss = 4046.47363568
Iteration 154, loss = 4021.47247081
Iteration 155, loss = 4004.38412541
Iteration 156, loss = 3980.59782634
Iteration 157, loss = 3949.93642108
Iteration 158, loss = 3927.98665023
Iteration 159, loss = 3904.26121670
Iteration 160, loss = 3882.46912165
Iteration 161, loss = 3864.24636979
Iteration 162, loss = 3851.55879852
Iteration 163, loss = 3828.74276647
Iteration 164, loss = 3800.88511165
Iteration 165, loss = 3768.42282226
Iteration 166, loss = 3747.16556705
Iteration 167, loss = 3740.17578194
Iteration 168, loss = 3731.00614265
Iteration 169, loss = 3715.11770574
Iteration 170, loss = 3692.99172526
Iteration 171, loss = 3670.13133471
Iteration 172, loss = 3649.30298128
Iteration 173, loss = 3625.31217636
Iteration 174, loss = 3602.87304580
Iteration 175, loss = 3585.80548391
Iteration 176, loss = 3572.73626870
Iteration 177, loss = 3556.56412561
Iteration 178, loss = 3538.54085532
Iteration 179, loss = 3524.19606762
Iteration 180, loss = 3505.71401303
Iteration 181, loss = 3486.21476911
Iteration 182, loss = 3472.94021644
Iteration 183, loss = 3459.03293685
Iteration 184, loss = 3443.33477704
Iteration 185, loss = 3428.16509272
Iteration 186, loss = 3413.85709549
Iteration 187, loss = 3397.95315640
Iteration 188, loss = 3384.85422744
Iteration 189, loss = 3369.02885863
Iteration 190, loss = 3355.14118498
Iteration 191, loss = 3345.32986504
Iteration 192, loss = 3329.50666086
Iteration 193, loss = 3315.73578882
Iteration 194, loss = 3304.02288611
Iteration 195, loss = 3292.59533864
Iteration 196, loss = 3285.52764169
Iteration 197, loss = 3272.32500834
Iteration 198, loss = 3256.06229563
Iteration 199, loss = 3241.73478022
Iteration 200, loss = 3226.75950837
Iteration 201, loss = 3212.83078153
Iteration 202, loss = 3200.43282774
Iteration 203, loss = 3194.99557337
Iteration 204, loss = 3186.33838212
Iteration 205, loss = 3178.22208802
Iteration 206, loss = 3166.13598208
Iteration 207, loss = 3151.31049178
Iteration 208, loss = 3139.54600098
Iteration 209, loss = 3129.38065695
Iteration 210, loss = 3117.34025226
Iteration 211, loss = 3104.56955983
Iteration 212, loss = 3094.49564727
Iteration 213, loss = 3079.92378181
Iteration 214, loss = 3072.42997798
Iteration 215, loss = 3061.03691605
Iteration 216, loss = 3050.56662508
Iteration 217, loss = 3041.43201463
Iteration 218, loss = 3033.71399903
Iteration 219, loss = 3021.87265008
Iteration 220, loss = 3019.24468528
Iteration 221, loss = 3021.89530517
Iteration 222, loss = 3013.87055315
Iteration 223, loss = 2998.51626437
Iteration 224, loss = 2983.25447147
Iteration 225, loss = 2965.74704897
Iteration 226, loss = 2949.00579628
Iteration 227, loss = 2941.75979934
Iteration 228, loss = 2931.27911956
Iteration 229, loss = 2920.30098467
Iteration 230, loss = 2912.36214907
Iteration 231, loss = 2906.91297899
Iteration 232, loss = 2899.38124847
Iteration 233, loss = 2888.99185234
Iteration 234, loss = 2876.54678532
Iteration 235, loss = 2864.05255744
Iteration 236, loss = 2857.24771792
Iteration 237, loss = 2845.90127620
Iteration 238, loss = 2835.90101418
Iteration 239, loss = 2830.91018500
Iteration 240, loss = 2830.40764093
Iteration 241, loss = 2825.50592590
Iteration 242, loss = 2818.84003674
Iteration 243, loss = 2809.51227019
Iteration 244, loss = 2796.00288942
Iteration 245, loss = 2778.57854218
Iteration 246, loss = 2768.08158643
Iteration 247, loss = 2760.97122049
Iteration 248, loss = 2750.88156861
Iteration 249, loss = 2742.39346099
Iteration 250, loss = 2733.14718178
Iteration 251, loss = 2726.63109921
Iteration 252, loss = 2722.13993575
Iteration 253, loss = 2714.80328556
Iteration 254, loss = 2703.88049727
Iteration 255, loss = 2691.72976862
Iteration 256, loss = 2686.79461120
Iteration 257, loss = 2681.65185098
Iteration 258, loss = 2674.97030386
Iteration 259, loss = 2666.21177199
Iteration 260, loss = 2658.84257040
Iteration 261, loss = 2653.26185443
Iteration 262, loss = 2641.24354418
Iteration 263, loss = 2629.54342608
Iteration 264, loss = 2618.69169705
Iteration 265, loss = 2612.44600295
Iteration 266, loss = 2614.64998888
Iteration 267, loss = 2610.37861788
Iteration 268, loss = 2600.50619995
Iteration 269, loss = 2583.90815610
Iteration 270, loss = 2573.47292177
Iteration 271, loss = 2571.94348050
Iteration 272, loss = 2566.10747635
Iteration 273, loss = 2561.71058049
Iteration 274, loss = 2558.00885678
Iteration 275, loss = 2552.57251769
Iteration 276, loss = 2543.36867185
Iteration 277, loss = 2532.80776471
Iteration 278, loss = 2525.58636876
Iteration 279, loss = 2516.65945961
Iteration 280, loss = 2506.47837932
Iteration 281, loss = 2500.03904621
Iteration 282, loss = 2493.21181422
Iteration 283, loss = 2487.89186241
Iteration 284, loss = 2479.18818001
Iteration 285, loss = 2474.25128098
Iteration 286, loss = 2471.06419619
Iteration 287, loss = 2462.82453262
Iteration 288, loss = 2454.81608015
Iteration 289, loss = 2449.71126759
Iteration 290, loss = 2446.78182060
Iteration 291, loss = 2440.98617479
Iteration 292, loss = 2443.68714761
Iteration 293, loss = 2443.20362912
Iteration 294, loss = 2440.49194868
Iteration 295, loss = 2439.90683372
Iteration 296, loss = 2435.18090171
Iteration 297, loss = 2432.70691234
Iteration 298, loss = 2429.62967175
Iteration 299, loss = 2424.94128156
Iteration 300, loss = 2419.02607429
Iteration 301, loss = 2404.32499401
Iteration 302, loss = 2391.36746200
Iteration 303, loss = 2391.54682608
Iteration 304, loss = 2380.29963327
Iteration 305, loss = 2370.66541563
Iteration 306, loss = 2361.19129042
Iteration 307, loss = 2359.85170297
Iteration 308, loss = 2352.67866663
Iteration 309, loss = 2344.19369582
Iteration 310, loss = 2334.50523030
Iteration 311, loss = 2329.26432238
Iteration 312, loss = 2323.44255359
Iteration 313, loss = 2317.24617120
Iteration 314, loss = 2313.78992142
Iteration 315, loss = 2306.81682623
Iteration 316, loss = 2299.80607590
Iteration 317, loss = 2294.52342040
Iteration 318, loss = 2292.73498272
Iteration 319, loss = 2285.46716690
Iteration 320, loss = 2280.48952726
Iteration 321, loss = 2275.12293891
Iteration 322, loss = 2270.99127633
Iteration 323, loss = 2272.46530019
Iteration 324, loss = 2276.25587403
Iteration 325, loss = 2284.10190656
Iteration 326, loss = 2267.96825439
Iteration 327, loss = 2249.61636464
Iteration 328, loss = 2246.55589673
Iteration 329, loss = 2236.46740342
Iteration 330, loss = 2228.63623661
Iteration 331, loss = 2223.53382610
Iteration 332, loss = 2218.68531079
Iteration 333, loss = 2219.12621476
Iteration 334, loss = 2211.17063460
Iteration 335, loss = 2201.75955423
Iteration 336, loss = 2198.59959984
Iteration 337, loss = 2201.90489389
Iteration 338, loss = 2202.53126751
Iteration 339, loss = 2201.36741403
Iteration 340, loss = 2203.84345487
Iteration 341, loss = 2197.06516374
Iteration 342, loss = 2186.50898805
Iteration 343, loss = 2172.22759553
Iteration 344, loss = 2162.74354381
Iteration 345, loss = 2154.83601680
Iteration 346, loss = 2147.06190663
Iteration 347, loss = 2150.97924182
Iteration 348, loss = 2138.76448102
Iteration 349, loss = 2135.43842352
Iteration 350, loss = 2139.51244598
Iteration 351, loss = 2139.23215008
Iteration 352, loss = 2126.89399118
Iteration 353, loss = 2115.78775640
Iteration 354, loss = 2117.09611003
Iteration 355, loss = 2115.46958859
Iteration 356, loss = 2111.64401063
Iteration 357, loss = 2105.24614140
Iteration 358, loss = 2097.90266753
Iteration 359, loss = 2096.29705686
Iteration 360, loss = 2094.14928327
Iteration 361, loss = 2091.64975736
Iteration 362, loss = 2093.80814613
Iteration 363, loss = 2090.76644822
Iteration 364, loss = 2076.13488594
Iteration 365, loss = 2074.32329101
Iteration 366, loss = 2086.12069976
Iteration 367, loss = 2091.48756950
Iteration 368, loss = 2096.09739769
Iteration 369, loss = 2091.17673692
Iteration 370, loss = 2073.06564943
Iteration 371, loss = 2063.27064917
Iteration 372, loss = 2051.18906225
Iteration 373, loss = 2051.25466424
Iteration 374, loss = 2047.71547286
Iteration 375, loss = 2042.47775945
Iteration 376, loss = 2031.99747211
Iteration 377, loss = 2035.72601067
Iteration 378, loss = 2030.26569482
Iteration 379, loss = 2020.51670990
Iteration 380, loss = 2015.15865486
Iteration 381, loss = 2017.14348439
Iteration 382, loss = 2015.09754919
Iteration 383, loss = 2008.27251671
Iteration 384, loss = 2001.72015598
Iteration 385, loss = 2001.79022429
Iteration 386, loss = 1993.34143871
Iteration 387, loss = 1989.64209925
Iteration 388, loss = 1988.62647753
Iteration 389, loss = 1993.13218041
Iteration 390, loss = 1993.07863501
Iteration 391, loss = 1995.22570554
Iteration 392, loss = 2002.49661293
Iteration 393, loss = 2010.82184518
Iteration 394, loss = 2008.54061565
Iteration 395, loss = 1991.91108272
Iteration 396, loss = 1976.77667079
Iteration 397, loss = 1969.54644490
Iteration 398, loss = 1971.43170517
Iteration 399, loss = 1978.67517290
Iteration 400, loss = 1982.64208478
Iteration 401, loss = 1977.97362551
Iteration 402, loss = 1961.40899789
Iteration 403, loss = 1957.11673254
Iteration 404, loss = 1953.58053472
Iteration 405, loss = 1955.25344997
Iteration 406, loss = 1954.96904268
Iteration 407, loss = 1967.90133291
Iteration 408, loss = 1979.20948592
Iteration 409, loss = 1974.11835792
Iteration 410, loss = 1961.45902992
Iteration 411, loss = 1944.84470145
Iteration 412, loss = 1927.86753095
Iteration 413, loss = 1915.33213034
Iteration 414, loss = 1920.49275656
Iteration 415, loss = 1931.51659438
Iteration 416, loss = 1928.03799281
Iteration 417, loss = 1910.59868097
Iteration 418, loss = 1894.94952987
Iteration 419, loss = 1888.65235507
Iteration 420, loss = 1890.94286777
Iteration 421, loss = 1886.01124523
Iteration 422, loss = 1882.38431913
Iteration 423, loss = 1879.35231134
Iteration 424, loss = 1877.06114410
Iteration 425, loss = 1874.95174544
Iteration 426, loss = 1873.35158337
Iteration 427, loss = 1870.99528839
Iteration 428, loss = 1864.73717599
Iteration 429, loss = 1860.39387011
Iteration 430, loss = 1859.26106531
Iteration 431, loss = 1859.35627006
Iteration 432, loss = 1856.94553840
Iteration 433, loss = 1851.28883770
Iteration 434, loss = 1845.80512395
Iteration 435, loss = 1842.07513189
Iteration 436, loss = 1838.65557967
Iteration 437, loss = 1835.88877716
Iteration 438, loss = 1831.66421166
Iteration 439, loss = 1828.28432636
Iteration 440, loss = 1826.52669927
Iteration 441, loss = 1824.15556281
Iteration 442, loss = 1822.87557775
Iteration 443, loss = 1821.99539080
Iteration 444, loss = 1821.03638715
Iteration 445, loss = 1818.54906433
Iteration 446, loss = 1811.75746310
Iteration 447, loss = 1813.69060407
Iteration 448, loss = 1818.19181344
Iteration 449, loss = 1818.60434733
Iteration 450, loss = 1814.47469149
Iteration 451, loss = 1812.22497861
Iteration 452, loss = 1810.96220893
Iteration 453, loss = 1806.63573964
Iteration 454, loss = 1800.26419550
Iteration 455, loss = 1793.31072143
Iteration 456, loss = 1788.39762967
Iteration 457, loss = 1786.49995068
Iteration 458, loss = 1786.85305966
Iteration 459, loss = 1788.90623833
Iteration 460, loss = 1789.55768168
Iteration 461, loss = 1784.91881271
Iteration 462, loss = 1781.05745869
Iteration 463, loss = 1777.14749255
Iteration 464, loss = 1773.64753329
Iteration 465, loss = 1768.27853160
Iteration 466, loss = 1766.39380911
Iteration 467, loss = 1766.10424883
Iteration 468, loss = 1767.30153061
Iteration 469, loss = 1763.63954579
Iteration 470, loss = 1757.16783514
Iteration 471, loss = 1762.89752042
Iteration 472, loss = 1762.98489970
Iteration 473, loss = 1758.41453135
Iteration 474, loss = 1752.70330468
Iteration 475, loss = 1744.63838148
Iteration 476, loss = 1743.15323324
Iteration 477, loss = 1746.42891063
Iteration 478, loss = 1749.17946682
Iteration 479, loss = 1749.85267067
Iteration 480, loss = 1746.37753770
Iteration 481, loss = 1740.20067063
Iteration 482, loss = 1736.58632759
Iteration 483, loss = 1727.21496913
Iteration 484, loss = 1724.92141360
Iteration 485, loss = 1730.01075829
Iteration 486, loss = 1742.64074845
Iteration 487, loss = 1732.54195104
Iteration 488, loss = 1719.81713719
Iteration 489, loss = 1712.10974217
Iteration 490, loss = 1708.82469221
Iteration 491, loss = 1709.26612857
Iteration 492, loss = 1709.09973787
Iteration 493, loss = 1707.77742081
Iteration 494, loss = 1709.40680200
Iteration 495, loss = 1704.99617648
Iteration 496, loss = 1699.13396813
Iteration 497, loss = 1695.45769379
Iteration 498, loss = 1694.19043508
Iteration 499, loss = 1690.97026380
Iteration 500, loss = 1688.69530278
Iteration 501, loss = 1690.53226672
Iteration 502, loss = 1690.13742151
Iteration 503, loss = 1693.86020064
Iteration 504, loss = 1688.92827264
Iteration 505, loss = 1686.83629761
Iteration 506, loss = 1679.21571439
Iteration 507, loss = 1674.26454254
Iteration 508, loss = 1670.68791527
Iteration 509, loss = 1668.28198567
Iteration 510, loss = 1663.40821159
Iteration 511, loss = 1666.27700713
Iteration 512, loss = 1666.20673971
Iteration 513, loss = 1663.84334356
Iteration 514, loss = 1658.79513992
Iteration 515, loss = 1656.76017173
Iteration 516, loss = 1649.71607012
Iteration 517, loss = 1649.09394670
Iteration 518, loss = 1652.96880411
Iteration 519, loss = 1660.71079018
Iteration 520, loss = 1666.11310916
Iteration 521, loss = 1663.23805584
Iteration 522, loss = 1658.36187395
Iteration 523, loss = 1655.84859653
Iteration 524, loss = 1660.65372176
Iteration 525, loss = 1660.89698473
Iteration 526, loss = 1657.26790280
Iteration 527, loss = 1648.30895702
Iteration 528, loss = 1641.03082707
Iteration 529, loss = 1631.96725489
Iteration 530, loss = 1623.63086351
Iteration 531, loss = 1623.36053857
Iteration 532, loss = 1628.82549738
Iteration 533, loss = 1633.31304994
Iteration 534, loss = 1629.76712764
Iteration 535, loss = 1623.06515086
Iteration 536, loss = 1616.59711742
Iteration 537, loss = 1624.86281222
Iteration 538, loss = 1641.04187454
Iteration 539, loss = 1649.90356602
Iteration 540, loss = 1642.65832690
Iteration 541, loss = 1628.35334487
Iteration 542, loss = 1611.77861228
Iteration 543, loss = 1604.00656412
Iteration 544, loss = 1602.10984082
Iteration 545, loss = 1611.00897020
Iteration 546, loss = 1610.09894127
Iteration 547, loss = 1605.90583833
Iteration 548, loss = 1603.02106160
Iteration 549, loss = 1600.58374962
Iteration 550, loss = 1592.89916319
Iteration 551, loss = 1589.53120724
Iteration 552, loss = 1585.89530627
Iteration 553, loss = 1583.59160450
Iteration 554, loss = 1583.79974717
Iteration 555, loss = 1582.79147881
Iteration 556, loss = 1579.30262265
Iteration 557, loss = 1578.28594673
Iteration 558, loss = 1573.96469795
Iteration 559, loss = 1570.74487635
Iteration 560, loss = 1566.63681522
Iteration 561, loss = 1571.00098015
Iteration 562, loss = 1572.71909097
Iteration 563, loss = 1568.51189994
Iteration 564, loss = 1561.53313466
Iteration 565, loss = 1558.83455654
Iteration 566, loss = 1556.38883594
Iteration 567, loss = 1553.25063550
Iteration 568, loss = 1550.62681186
Iteration 569, loss = 1548.91200993
Iteration 570, loss = 1546.93876661
Iteration 571, loss = 1543.75196118
Iteration 572, loss = 1544.67102558
Iteration 573, loss = 1541.21538010
Iteration 574, loss = 1542.20070163
Iteration 575, loss = 1544.34896894
Iteration 576, loss = 1545.65416518
Iteration 577, loss = 1540.59350862
Iteration 578, loss = 1533.90066741
Iteration 579, loss = 1532.41637527
Iteration 580, loss = 1531.68856109
Iteration 581, loss = 1529.58103918
Iteration 582, loss = 1526.97224913
Iteration 583, loss = 1527.51478490
Iteration 584, loss = 1529.12418869
Iteration 585, loss = 1525.81966591
Iteration 586, loss = 1518.39956930
Iteration 587, loss = 1516.17020776
Iteration 588, loss = 1518.10933823
Iteration 589, loss = 1520.58671744
Iteration 590, loss = 1519.43506833
Iteration 591, loss = 1519.53519789
Iteration 592, loss = 1520.93376753
Iteration 593, loss = 1515.70027978
Iteration 594, loss = 1514.59741519
Iteration 595, loss = 1503.96881364
Iteration 596, loss = 1500.91479250
Iteration 597, loss = 1513.02582721
Iteration 598, loss = 1508.75917801
Iteration 599, loss = 1499.13128144
Iteration 600, loss = 1490.25700376
Iteration 601, loss = 1493.71862531
Iteration 602, loss = 1495.20220777
Iteration 603, loss = 1491.35026352
Iteration 604, loss = 1487.48086170
Iteration 605, loss = 1482.35813404
Iteration 606, loss = 1481.79996183
Iteration 607, loss = 1478.40635597
Iteration 608, loss = 1474.76581004
Iteration 609, loss = 1474.29379088
Iteration 610, loss = 1472.99896149
Iteration 611, loss = 1469.86380563
Iteration 612, loss = 1466.60611284
Iteration 613, loss = 1466.50866156
Iteration 614, loss = 1464.69620052
Iteration 615, loss = 1470.69787524
Iteration 616, loss = 1477.52989422
Iteration 617, loss = 1473.89334760
Iteration 618, loss = 1464.00313484
Iteration 619, loss = 1458.00493381
Iteration 620, loss = 1454.96380264
Iteration 621, loss = 1456.77430195
Iteration 622, loss = 1452.85945230
Iteration 623, loss = 1448.53349668
Iteration 624, loss = 1449.16847731
Iteration 625, loss = 1447.73550266
Iteration 626, loss = 1446.70255657
Iteration 627, loss = 1443.46463033
Iteration 628, loss = 1440.79189941
Iteration 629, loss = 1439.18155676
Iteration 630, loss = 1437.20896885
Iteration 631, loss = 1438.92577953
Iteration 632, loss = 1440.42902429
Iteration 633, loss = 1436.85299560
Iteration 634, loss = 1430.25531847
Iteration 635, loss = 1428.70384434
Iteration 636, loss = 1428.17216277
Iteration 637, loss = 1427.78120073
Iteration 638, loss = 1427.06632482
Iteration 639, loss = 1423.85760239
Iteration 640, loss = 1426.65215193
Iteration 641, loss = 1426.78712902
Iteration 642, loss = 1425.64046612
Iteration 643, loss = 1420.59761706
Iteration 644, loss = 1419.34804801
Iteration 645, loss = 1416.44578426
Iteration 646, loss = 1414.27460098
Iteration 647, loss = 1413.79840856
Iteration 648, loss = 1413.09093289
Iteration 649, loss = 1416.42492065
Iteration 650, loss = 1417.97762446
Iteration 651, loss = 1414.99338107
Iteration 652, loss = 1409.21534990
Iteration 653, loss = 1401.28481592
Iteration 654, loss = 1394.92791853
Iteration 655, loss = 1402.84799253
Iteration 656, loss = 1409.91493498
Iteration 657, loss = 1409.60719119
Iteration 658, loss = 1413.64723779
Iteration 659, loss = 1415.29855366
Iteration 660, loss = 1410.81922570
Iteration 661, loss = 1410.59635366
Iteration 662, loss = 1411.91945466
Iteration 663, loss = 1409.32437176
Iteration 664, loss = 1399.14452850
Iteration 665, loss = 1390.79195764
Iteration 666, loss = 1383.77203982
Iteration 667, loss = 1382.95805546
Iteration 668, loss = 1382.83932764
Iteration 669, loss = 1378.81124682
Iteration 670, loss = 1380.65155437
Iteration 671, loss = 1381.87946461
Iteration 672, loss = 1378.50741570
Iteration 673, loss = 1375.38531399
Iteration 674, loss = 1374.91731310
Iteration 675, loss = 1376.40944282
Iteration 676, loss = 1376.09348117
Iteration 677, loss = 1373.07622893
Iteration 678, loss = 1369.42168542
Iteration 679, loss = 1366.22094957
Iteration 680, loss = 1361.06999375
Iteration 681, loss = 1360.06831825
Iteration 682, loss = 1364.66906763
Iteration 683, loss = 1364.17741088
Iteration 684, loss = 1363.35098184
Iteration 685, loss = 1359.42674078
Iteration 686, loss = 1352.91060961
Iteration 687, loss = 1349.42053211
Iteration 688, loss = 1352.92605442
Iteration 689, loss = 1357.88668235
Iteration 690, loss = 1363.53040555
Iteration 691, loss = 1369.39154134
Iteration 692, loss = 1363.31483595
Iteration 693, loss = 1360.90276410
Iteration 694, loss = 1358.59604766
Iteration 695, loss = 1355.98275966
Iteration 696, loss = 1353.88092889
Iteration 697, loss = 1352.85550896
Iteration 698, loss = 1348.05919273
Iteration 699, loss = 1352.27092163
Iteration 700, loss = 1349.15348987
Iteration 701, loss = 1342.73967868
Iteration 702, loss = 1339.51879452
Iteration 703, loss = 1335.38715967
Iteration 704, loss = 1330.90369009
Iteration 705, loss = 1328.73519484
Iteration 706, loss = 1328.62955337
Iteration 707, loss = 1326.23183122
Iteration 708, loss = 1331.89531392
Iteration 709, loss = 1336.18881188
Iteration 710, loss = 1331.26015550
Iteration 711, loss = 1329.13241910
Iteration 712, loss = 1326.63800476
Iteration 713, loss = 1321.64552091
Iteration 714, loss = 1318.74284219
Iteration 715, loss = 1317.19362099
Iteration 716, loss = 1312.25633301
Iteration 717, loss = 1309.73801404
Iteration 718, loss = 1308.69965250
Iteration 719, loss = 1304.44968108
Iteration 720, loss = 1299.18172199
Iteration 721, loss = 1298.74557953
Iteration 722, loss = 1305.78491670
Iteration 723, loss = 1303.37834517
Iteration 724, loss = 1297.81636470
Iteration 725, loss = 1293.74256827
Iteration 726, loss = 1293.12376272
Iteration 727, loss = 1291.69458181
Iteration 728, loss = 1291.80545581
Iteration 729, loss = 1293.26937457
Iteration 730, loss = 1290.64063992
Iteration 731, loss = 1285.15880908
Iteration 732, loss = 1286.18697369
Iteration 733, loss = 1290.02663503
Iteration 734, loss = 1291.91723008
Iteration 735, loss = 1293.51208553
Iteration 736, loss = 1288.84711609
Iteration 737, loss = 1278.58279102
Iteration 738, loss = 1271.82697853
Iteration 739, loss = 1271.20748466
Iteration 740, loss = 1268.58188214
Iteration 741, loss = 1268.42115307
Iteration 742, loss = 1267.15725979
Iteration 743, loss = 1264.05278805
Iteration 744, loss = 1265.48491512
Iteration 745, loss = 1270.83936613
Iteration 746, loss = 1266.45654299
Iteration 747, loss = 1255.56650569
Iteration 748, loss = 1252.81121777
Iteration 749, loss = 1268.02948023
Iteration 750, loss = 1270.29644993
Iteration 751, loss = 1258.37321884
Iteration 752, loss = 1247.72759467
Iteration 753, loss = 1247.19277986
Iteration 754, loss = 1256.31553629
Iteration 755, loss = 1257.94935819
Iteration 756, loss = 1254.03692471
Iteration 757, loss = 1243.68607439
Iteration 758, loss = 1245.32960136
Iteration 759, loss = 1256.62558813
Iteration 760, loss = 1254.49669121
Iteration 761, loss = 1240.77118291
Iteration 762, loss = 1249.03366098
Iteration 763, loss = 1247.63469329
Iteration 764, loss = 1250.16079380
Iteration 765, loss = 1249.91495009
Iteration 766, loss = 1242.20072934
Iteration 767, loss = 1239.29463933
Iteration 768, loss = 1246.51279363
Iteration 769, loss = 1246.66849893
Iteration 770, loss = 1236.94947107
Iteration 771, loss = 1232.24919917
Iteration 772, loss = 1225.24004232
Iteration 773, loss = 1220.94679580
Iteration 774, loss = 1224.51868353
Iteration 775, loss = 1243.85626134
Iteration 776, loss = 1237.56444692
Iteration 777, loss = 1221.04128379
Iteration 778, loss = 1220.39650653
Iteration 779, loss = 1230.95060312
Iteration 780, loss = 1240.56195712
Iteration 781, loss = 1240.05042540
Iteration 782, loss = 1227.29158400
Iteration 783, loss = 1214.31394180
Iteration 784, loss = 1209.67494218
Iteration 785, loss = 1206.83940870
Iteration 786, loss = 1205.70836128
Iteration 787, loss = 1208.17633719
Iteration 788, loss = 1209.64884098
Iteration 789, loss = 1208.71358396
Iteration 790, loss = 1203.14833560
Iteration 791, loss = 1194.06355589
Iteration 792, loss = 1193.86491200
Iteration 793, loss = 1203.99638228
Iteration 794, loss = 1202.10745337
Iteration 795, loss = 1191.70401844
Iteration 796, loss = 1191.66386504
Iteration 797, loss = 1188.41067056
Iteration 798, loss = 1186.25257778
Iteration 799, loss = 1184.98491027
Iteration 800, loss = 1182.30804401
Iteration 801, loss = 1180.88703220
Iteration 802, loss = 1175.40114148
Iteration 803, loss = 1181.69442451
Iteration 804, loss = 1184.24633039
Iteration 805, loss = 1179.42967666
Iteration 806, loss = 1172.09739421
Iteration 807, loss = 1169.88622676
Iteration 808, loss = 1172.46777380
Iteration 809, loss = 1178.62754927
Iteration 810, loss = 1185.76235283
Iteration 811, loss = 1181.70115398
Iteration 812, loss = 1171.15990269
Iteration 813, loss = 1163.31123428
Iteration 814, loss = 1161.57885370
Iteration 815, loss = 1167.15583579
Iteration 816, loss = 1172.83255095
Iteration 817, loss = 1172.06853225
Iteration 818, loss = 1169.25686407
Iteration 819, loss = 1164.44537133
Iteration 820, loss = 1154.25101650
Iteration 821, loss = 1156.18760955
Iteration 822, loss = 1165.47657502
Iteration 823, loss = 1169.35981551
Iteration 824, loss = 1162.32710993
Iteration 825, loss = 1150.77552734
Iteration 826, loss = 1147.36611787
Iteration 827, loss = 1151.74744683
Iteration 828, loss = 1161.94597929
Iteration 829, loss = 1177.75776409
Iteration 830, loss = 1176.13546004
Iteration 831, loss = 1155.23872452
Iteration 832, loss = 1142.09179096
Iteration 833, loss = 1158.55404084
Iteration 834, loss = 1168.26252615
Iteration 835, loss = 1161.80214824
Iteration 836, loss = 1144.79504768
Iteration 837, loss = 1134.05572940
Iteration 838, loss = 1134.66935325
Iteration 839, loss = 1135.48172281
Iteration 840, loss = 1136.53184981
Iteration 841, loss = 1139.89840804
Iteration 842, loss = 1132.12648827
Iteration 843, loss = 1138.58055115
Iteration 844, loss = 1135.15288708
Iteration 845, loss = 1130.34798262
Iteration 846, loss = 1130.98343952
Iteration 847, loss = 1132.47527192
Iteration 848, loss = 1131.92883866
Iteration 849, loss = 1129.60702466
Iteration 850, loss = 1120.88029918
Iteration 851, loss = 1125.29607314
Iteration 852, loss = 1132.40243190
Iteration 853, loss = 1133.08145353
Iteration 854, loss = 1121.60977898
Iteration 855, loss = 1109.73667935
Iteration 856, loss = 1115.83971122
Iteration 857, loss = 1136.78963183
Iteration 858, loss = 1135.34491856
Iteration 859, loss = 1118.34239896
Iteration 860, loss = 1109.72120378
Iteration 861, loss = 1111.48655199
Iteration 862, loss = 1112.04178097
Iteration 863, loss = 1108.51310633
Iteration 864, loss = 1109.61418101
Iteration 865, loss = 1115.86116467
Iteration 866, loss = 1120.59185413
Iteration 867, loss = 1108.72984236
Iteration 868, loss = 1096.09162616
Iteration 869, loss = 1094.27846590
Iteration 870, loss = 1101.31179789
Iteration 871, loss = 1103.05935387
Iteration 872, loss = 1095.85329100
Iteration 873, loss = 1088.27982005
Iteration 874, loss = 1088.43023962
Iteration 875, loss = 1091.61046325
Iteration 876, loss = 1088.66284017
Iteration 877, loss = 1087.12072345
Iteration 878, loss = 1092.30667331
Iteration 879, loss = 1093.17456807
Iteration 880, loss = 1084.64776767
Iteration 881, loss = 1081.04029809
Iteration 882, loss = 1078.96087619
Iteration 883, loss = 1075.27671467
Iteration 884, loss = 1077.19185773
Iteration 885, loss = 1085.37737885
Iteration 886, loss = 1076.05979957
Iteration 887, loss = 1083.57246697
Iteration 888, loss = 1085.55054721
Iteration 889, loss = 1071.58310386
Iteration 890, loss = 1064.11384560
Iteration 891, loss = 1069.69646121
Iteration 892, loss = 1080.02035616
Iteration 893, loss = 1078.65873618
Iteration 894, loss = 1068.46135246
Iteration 895, loss = 1061.94676070
Iteration 896, loss = 1060.59432486
Iteration 897, loss = 1059.55664382
Iteration 898, loss = 1056.02678102
Iteration 899, loss = 1057.00340073
Iteration 900, loss = 1061.95486330
Iteration 901, loss = 1058.78442391
Iteration 902, loss = 1052.33274676
Iteration 903, loss = 1049.95972918
Iteration 904, loss = 1048.45997505
Iteration 905, loss = 1046.25224625
Iteration 906, loss = 1047.01364768
Iteration 907, loss = 1045.92716952
Iteration 908, loss = 1048.65428096
Iteration 909, loss = 1043.89424978
Iteration 910, loss = 1048.18131807
Iteration 911, loss = 1050.70879396
Iteration 912, loss = 1048.50688084
Iteration 913, loss = 1044.72323648
Iteration 914, loss = 1036.80161393
Iteration 915, loss = 1036.17035103
Iteration 916, loss = 1037.33912916
Iteration 917, loss = 1037.09671820
Iteration 918, loss = 1034.14808401
Iteration 919, loss = 1031.63927099
Iteration 920, loss = 1036.58734834
Iteration 921, loss = 1040.35848808
Iteration 922, loss = 1032.63820816
Iteration 923, loss = 1024.39893249
Iteration 924, loss = 1031.93591240
Iteration 925, loss = 1034.25428376
Iteration 926, loss = 1037.39783200
Iteration 927, loss = 1039.30747942
Iteration 928, loss = 1030.48314173
Iteration 929, loss = 1021.82881285
Iteration 930, loss = 1021.18303352
Iteration 931, loss = 1025.95320109
Iteration 932, loss = 1024.57598775
Iteration 933, loss = 1017.55940276
Iteration 934, loss = 1017.73520476
Iteration 935, loss = 1027.42511456
Iteration 936, loss = 1039.20795476
Iteration 937, loss = 1031.45954819
Iteration 938, loss = 1012.36754372
Iteration 939, loss = 1008.63123810
Iteration 940, loss = 1014.23109663
Iteration 941, loss = 1016.26508538
Iteration 942, loss = 1015.29070852
Iteration 943, loss = 1015.98893726
Iteration 944, loss = 1018.68669706
Iteration 945, loss = 1010.22530984
Iteration 946, loss = 1003.93166422
Iteration 947, loss = 1007.47871429
Iteration 948, loss = 1015.59762812
Iteration 949, loss = 1022.16366216
Iteration 950, loss = 1008.88645525
Iteration 951, loss = 993.87000775
Iteration 952, loss = 1001.39604134
Iteration 953, loss = 1006.42503360
Iteration 954, loss = 1004.80576067
Iteration 955, loss = 997.07099722
Iteration 956, loss = 988.64201818
Iteration 957, loss = 989.77021278
Iteration 958, loss = 999.29411739
Iteration 959, loss = 1003.52944283
Iteration 960, loss = 988.82201750
Iteration 961, loss = 978.83382132
Iteration 962, loss = 987.74766590
Iteration 963, loss = 987.04326963
Iteration 964, loss = 983.78628760
Iteration 965, loss = 976.57557935
Iteration 966, loss = 972.41193326
Iteration 967, loss = 973.74227602
Iteration 968, loss = 975.21302952
Iteration 969, loss = 975.62476010
Iteration 970, loss = 978.87801172
Iteration 971, loss = 979.24366694
Iteration 972, loss = 970.44305125
Iteration 973, loss = 965.94989394
Iteration 974, loss = 969.42679223
Iteration 975, loss = 969.56444459
Iteration 976, loss = 968.12606414
Iteration 977, loss = 966.71649121
Iteration 978, loss = 968.10049920
Iteration 979, loss = 972.20792771
Iteration 980, loss = 971.15839377
Iteration 981, loss = 964.79726118
Iteration 982, loss = 962.66356745
Iteration 983, loss = 963.29915339
Iteration 984, loss = 962.20666800
Iteration 985, loss = 956.76160737
Iteration 986, loss = 950.40293771
Iteration 987, loss = 948.35718320
Iteration 988, loss = 949.11139985
Iteration 989, loss = 949.98044925
Iteration 990, loss = 946.58452277
Iteration 991, loss = 949.41431573
Iteration 992, loss = 950.13044503
Iteration 993, loss = 949.85769819
Iteration 994, loss = 948.53983802
Iteration 995, loss = 942.26790838
Iteration 996, loss = 936.41131691
Iteration 997, loss = 935.13943048
Iteration 998, loss = 945.14416900
Iteration 999, loss = 951.51797395
Iteration 1000, loss = 946.78529291
Iteration 1001, loss = 933.25010563
Iteration 1002, loss = 933.46678734
Iteration 1003, loss = 938.83332857
Iteration 1004, loss = 933.34217928
Iteration 1005, loss = 924.40402516
Iteration 1006, loss = 921.25999135
Iteration 1007, loss = 919.83971397
Iteration 1008, loss = 925.25315566
Iteration 1009, loss = 932.25411494
Iteration 1010, loss = 934.75756744
Iteration 1011, loss = 926.37109276
Iteration 1012, loss = 917.65032704
Iteration 1013, loss = 918.47592478
Iteration 1014, loss = 923.42051350
Iteration 1015, loss = 925.96505235
Iteration 1016, loss = 925.97057790
Iteration 1017, loss = 920.93135994
Iteration 1018, loss = 918.69002763
Iteration 1019, loss = 928.37499610
Iteration 1020, loss = 929.09884218
Iteration 1021, loss = 917.67828512
Iteration 1022, loss = 910.69688029
Iteration 1023, loss = 908.28759761
Iteration 1024, loss = 909.07777054
Iteration 1025, loss = 908.41641378
Iteration 1026, loss = 906.96992353
Iteration 1027, loss = 903.23805905
Iteration 1028, loss = 900.74209434
Iteration 1029, loss = 907.98242919
Iteration 1030, loss = 915.13584861
Iteration 1031, loss = 910.60678542
Iteration 1032, loss = 899.14757982
Iteration 1033, loss = 903.14706873
Iteration 1034, loss = 921.66381451
Iteration 1035, loss = 923.20986213
Iteration 1036, loss = 911.59483273
Iteration 1037, loss = 898.04472086
Iteration 1038, loss = 895.22243165
Iteration 1039, loss = 903.11189028
Iteration 1040, loss = 901.77688069
Iteration 1041, loss = 890.28440324
Iteration 1042, loss = 882.80966386
Iteration 1043, loss = 882.67321928
Iteration 1044, loss = 883.02270612
Iteration 1045, loss = 882.41185954
Iteration 1046, loss = 881.26732127
Iteration 1047, loss = 879.61862114
Iteration 1048, loss = 877.12081388
Iteration 1049, loss = 874.36317595
Iteration 1050, loss = 880.22318904
Iteration 1051, loss = 896.91045236
Iteration 1052, loss = 909.46083950
Iteration 1053, loss = 892.01393554
Iteration 1054, loss = 871.44753215
Iteration 1055, loss = 874.24222120
Iteration 1056, loss = 896.08408368
Iteration 1057, loss = 906.15508471
Iteration 1058, loss = 884.70562360
Iteration 1059, loss = 863.47943837
Iteration 1060, loss = 881.52743492
Iteration 1061, loss = 908.02394313
Iteration 1062, loss = 899.82278035
Iteration 1063, loss = 874.90173468
Iteration 1064, loss = 865.30501933
Iteration 1065, loss = 858.78929278
Iteration 1066, loss = 856.42570610
Iteration 1067, loss = 858.28182930
Iteration 1068, loss = 865.73680549
Iteration 1069, loss = 858.65584458
Iteration 1070, loss = 857.12286829
Iteration 1071, loss = 853.43189913
Iteration 1072, loss = 849.08929114
Iteration 1073, loss = 849.51898211
Iteration 1074, loss = 848.12376126
Iteration 1075, loss = 848.18777130
Iteration 1076, loss = 845.66276178
Iteration 1077, loss = 849.69686894
Iteration 1078, loss = 850.35411895
Iteration 1079, loss = 848.02393074
Iteration 1080, loss = 852.50838942
Iteration 1081, loss = 850.59133481
Iteration 1082, loss = 846.35859720
Iteration 1083, loss = 841.23334216
Iteration 1084, loss = 836.04347717
Iteration 1085, loss = 837.27930001
Iteration 1086, loss = 839.21036594
Iteration 1087, loss = 836.99025609
Iteration 1088, loss = 834.45407616
Iteration 1089, loss = 833.21850359
Iteration 1090, loss = 833.28410456
Iteration 1091, loss = 833.27929207
Iteration 1092, loss = 833.89886918
Iteration 1093, loss = 832.58172341
Iteration 1094, loss = 834.51816851
Iteration 1095, loss = 830.77235296
Iteration 1096, loss = 827.18126015
Iteration 1097, loss = 823.53741761
Iteration 1098, loss = 826.18327571
Iteration 1099, loss = 823.73341519
Iteration 1100, loss = 818.55218861
Iteration 1101, loss = 819.89984572
Iteration 1102, loss = 821.50752087
Iteration 1103, loss = 819.51937501
Iteration 1104, loss = 815.25031917
Iteration 1105, loss = 814.12289470
Iteration 1106, loss = 819.28030195
Iteration 1107, loss = 822.72227903
Iteration 1108, loss = 817.12315459
Iteration 1109, loss = 813.26424704
Iteration 1110, loss = 816.80931638
Iteration 1111, loss = 829.36786382
Iteration 1112, loss = 832.06813469
Iteration 1113, loss = 821.40919757
Iteration 1114, loss = 808.62191200
Iteration 1115, loss = 805.83703362
Iteration 1116, loss = 812.90857229
Iteration 1117, loss = 812.15340078
Iteration 1118, loss = 810.70186480
Iteration 1119, loss = 808.60204749
Iteration 1120, loss = 809.13287620
Iteration 1121, loss = 803.91242075
Iteration 1122, loss = 803.23052946
Iteration 1123, loss = 804.67749029
Iteration 1124, loss = 803.52074095
Iteration 1125, loss = 799.44258954
Iteration 1126, loss = 796.30188342
Iteration 1127, loss = 794.41887999
Iteration 1128, loss = 790.51418790
Iteration 1129, loss = 789.39520003
Iteration 1130, loss = 791.08592021
Iteration 1131, loss = 794.05684446
Iteration 1132, loss = 791.81993917
Iteration 1133, loss = 791.61938700
Iteration 1134, loss = 791.73518879
Iteration 1135, loss = 789.38819976
Iteration 1136, loss = 790.08669117
Iteration 1137, loss = 794.95444679
Iteration 1138, loss = 796.87599706
Iteration 1139, loss = 790.29158789
Iteration 1140, loss = 786.76135689
Iteration 1141, loss = 797.09423733
Iteration 1142, loss = 800.44265620
Iteration 1143, loss = 795.86404560
Iteration 1144, loss = 787.43119027
Iteration 1145, loss = 793.22237237
Iteration 1146, loss = 793.30039275
Iteration 1147, loss = 783.34577810
Iteration 1148, loss = 777.04620985
Iteration 1149, loss = 775.89997491
Iteration 1150, loss = 783.65504877
Iteration 1151, loss = 778.36681787
Iteration 1152, loss = 769.63918879
Iteration 1153, loss = 766.44791344
Iteration 1154, loss = 767.86132464
Iteration 1155, loss = 773.72056587
Iteration 1156, loss = 774.59482889
Iteration 1157, loss = 769.92555258
Iteration 1158, loss = 765.64063248
Iteration 1159, loss = 765.21044079
Iteration 1160, loss = 768.81486719
Iteration 1161, loss = 772.72258220
Iteration 1162, loss = 767.97424366
Iteration 1163, loss = 763.89216090
Iteration 1164, loss = 763.24035915
Iteration 1165, loss = 763.64490747
Iteration 1166, loss = 758.96064111
Iteration 1167, loss = 759.10775746
Iteration 1168, loss = 762.50414481
Iteration 1169, loss = 758.90560439
Iteration 1170, loss = 755.92458385
Iteration 1171, loss = 752.22037079
Iteration 1172, loss = 750.09989202
Iteration 1173, loss = 759.27607923
Iteration 1174, loss = 772.16793265
Iteration 1175, loss = 774.92427669
Iteration 1176, loss = 767.22000154
Iteration 1177, loss = 753.79948986
Iteration 1178, loss = 763.51072976
Iteration 1179, loss = 757.70870311
Iteration 1180, loss = 747.89139230
Iteration 1181, loss = 744.38593647
Iteration 1182, loss = 745.59145484
Iteration 1183, loss = 743.90868063
Iteration 1184, loss = 739.76214078
Iteration 1185, loss = 740.21971835
Iteration 1186, loss = 749.43106788
Iteration 1187, loss = 744.06760405
Iteration 1188, loss = 736.72487803
Iteration 1189, loss = 739.15347086
Iteration 1190, loss = 737.99620287
Iteration 1191, loss = 734.49444821
Iteration 1192, loss = 736.91580093
Iteration 1193, loss = 734.51080241
Iteration 1194, loss = 731.36493038
Iteration 1195, loss = 730.28645949
Iteration 1196, loss = 731.93805502
Iteration 1197, loss = 738.82163243
Iteration 1198, loss = 732.04096092
Iteration 1199, loss = 729.31698113
Iteration 1200, loss = 732.18905071
Iteration 1201, loss = 730.61309982
Iteration 1202, loss = 725.65470004
Iteration 1203, loss = 721.34189584
Iteration 1204, loss = 721.56361905
Iteration 1205, loss = 725.39266888
Iteration 1206, loss = 723.54943922
Iteration 1207, loss = 720.06957123
Iteration 1208, loss = 721.53816244
Iteration 1209, loss = 722.00828979
Iteration 1210, loss = 718.60485426
Iteration 1211, loss = 718.02775123
Iteration 1212, loss = 715.48802974
Iteration 1213, loss = 731.86142892
Iteration 1214, loss = 746.15937715
Iteration 1215, loss = 726.82241785
Iteration 1216, loss = 721.17668980
Iteration 1217, loss = 723.70109294
Iteration 1218, loss = 718.29546013
Iteration 1219, loss = 722.51728645
Iteration 1220, loss = 748.07505630
Iteration 1221, loss = 747.81424099
Iteration 1222, loss = 722.33474529
Iteration 1223, loss = 722.01547029
Training loss did not improve more than tol=0.000100 for 10 consecutive epochs.
Stopping.
```

```
[23]: MLPRegressor(alpha=0.1, hidden_layer_sizes=(1000, 500), max_iter=10000,
                   random_state=1, verbose=10)
```

5 4.3

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

      def print_metrics(y_test, y_pred, squared=False):
          print(f"R^2: {r2_score(y_test, y_pred)}")
          crit_name = "MSE" if squared else "RMSE"
          print(f"{crit_name}: {mean_squared_error(y_test, y_pred, squared=squared)}")
          print(f"MAE: {mean_absolute_error(y_test, y_pred)}")

      def show_metric_for(y_pred, n=365):
        print_metrics(scaler.inverse_transform(ins_y_test.reshape(-1, 1)).flatten()[:
       ↪n], y_pred[:n])
```

```
[26]: import gmdh
      gmdh_combi = gmdh.Combi()
```

```
gmdh_combi.fit(ins_X_train_scaled, ins_y_train, verbose=1, n_jobs=-1,␣
 ↪test_size=0.2, limit=0,
                       criterion=gmdh.Criterion(gmdh.CriterionType.REGULARITY))
```

```
LEVEL 1  [=========================] 100% [00m:00s] (5 combinations)
error=695146.7162
LEVEL 2  [=========================] 100% [00m:00s] (10 combinations)
error=670212.555515
LEVEL 3  [=========================] 100% [00m:00s] (10 combinations)
error=648557.585664
LEVEL 4  [=========================] 100% [00m:00s] (5 combinations)
error=659882.164746
```

[26]:
```
<gmdh.gmdh.Combi at 0x1b8b220cf40>
```

[27]:
```
gmdh_mia = gmdh.Mia()
gmdh_mia.fit(ins_X_train_scaled, ins_y_train, verbose=1, n_jobs=-1, test_size=0.
 ↪2, limit=0, k_best=9,
                     criterion=gmdh.Criterion(gmdh.CriterionType.
 ↪SYM_REGULARITY),
                     polynomial_type=gmdh.PolynomialType.LINEAR)
```

```
LEVEL 1  [=========================] 100% [00m:00s] (10 combinations)
error=3.67867e+06
LEVEL 2  [=========================] 100% [00m:00s] (36 combinations)
error=3.57343e+06
LEVEL 3  [=========================] 100% [00m:00s] (36 combinations)
error=3.52661e+06
LEVEL 4  [=========================] 100% [00m:00s] (36 combinations)
error=3.52347e+06
LEVEL 5  [=========================] 100% [00m:00s] (36 combinations)
error=3.50106e+06
LEVEL 6  [=========================] 100% [00m:00s] (36 combinations)
error=3.49533e+06
LEVEL 7  [=========================] 100% [00m:00s] (36 combinations)
error=3.49154e+06
LEVEL 8  [=========================] 100% [00m:00s] (36 combinations)
error=3.4945e+06
```

[27]:
```
<gmdh.gmdh.Mia at 0x1b8b63b7c10>
```

[28]:
```
pred_mia = gmdh_mia.predict(ins_X_test_scaled)
print_metrics(ins_y_test, pred_mia)
```

```
R^2: 0.13334188124042634
RMSE: 116.98091376329361
MAE: 89.65482934967766
```

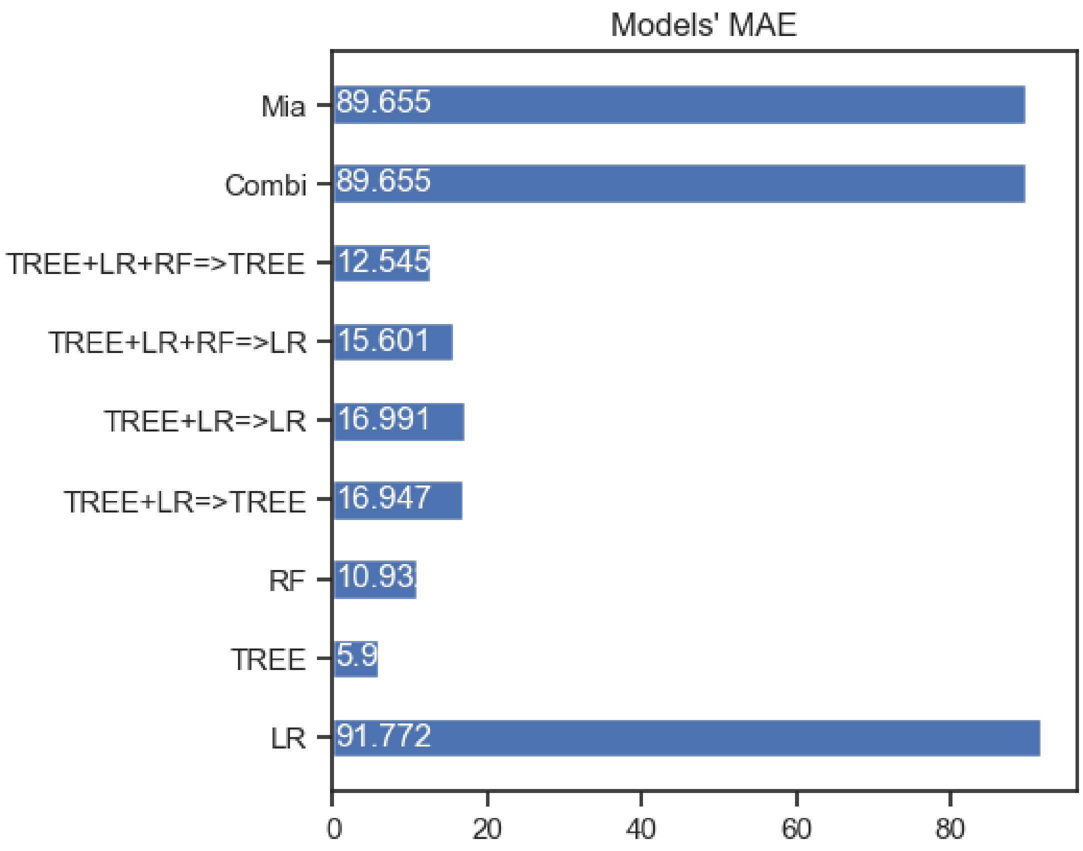
```
c:\Users\yulia\AppData\Local\Programs\Python\Python39\lib\site-
packages\sklearn\metrics\_regression.py:483: FutureWarning: 'squared' is
deprecated in version 1.4 and will be removed in 1.6. To calculate the root mean
squared error, use the function'root_mean_squared_error'.
  warnings.warn(
```

[31]:

```
def vis_models_quality(array_metric, array_labels, str_header, figsize=(8, 8)):
    fig, ax1 = plt.subplots(figsize=figsize)
    pos = np.arange(len(array_metric))
    rects = ax1.barh(pos, array_metric,
                     align='center',
                     height=0.5,
                     tick_label=array_labels)
    ax1.set_title(str_header)
    for a,b in zip(pos, array_metric):
        plt.text(0.2, a-0.1, str(round(b,3)), color='white')
    plt.show()
```

[45]:

```
dict_mae = {
            'LR': 91.77213251714615,
            'TREE': 5.932203389830509,
            'RF': 10.932203389830509,
            'TREE+LR=>TREE': 16.947463768115945,
            'TREE+LR=>LR': 16.991272889514768,
            'TREE+LR+RF=>LR': 15.601464066506784,
            'TREE+LR+RF=>TREE': 12.545289855072465,
            'Combi': 89.65482934967766,
            'Mia': 89.65482934967766,
            }
vis_models_quality(list(dict_mae.values()), list(dict_mae.keys()), 'Models\'␣
 ↪MAE')
```


Models' MAE
Mia
89.655
Combi
89.655
TREE+LR+RF=>TREE
12.545
TREE+LR+RF=>LR
15.601
TREE+LR=>LR
16.991
TREE+LR=>TREE
16.947
RF
10.93
TREE
5.9
LR
91.772
0
20
40
60
80