optuna-tuning

May 12, 2023

[ ]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np

from keras.layers import LSTM, Dense, Dropout
from sklearn.model_selection import TimeSeriesSplit
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression, LogisticRegression
from keras.models import Sequential, load_model
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split

train = pd.read_csv("train.csv")
testing = pd.read_csv("test.csv")
testing = testing[["Factor A", "Factor D", "Factor F", "Factor G", "Factor H"]]
test = testing.to_numpy()
one = train["TGD Consultancy Share price"]
two = train["TGD Automobiles Share price"]
three = train["TGD Power Share price"]
train = train[["dates", "Factor A", "Factor D", "Factor F", "Factor G", "Factor␣
 ↪H", "TGD Consultancy Share price", "TGD Automobiles Share price", "TGD Power␣
 ↪Share price"]]
display(train)
print(test)

features = train.drop(columns = ["dates", "TGD Consultancy Share price", "TGD␣
 ↪Automobiles Share price", "TGD Power Share price"], axis = 1).to_numpy()

target = train[["TGD Consultancy Share price", "TGD Automobiles Share price",␣
 ↪"TGD Power Share price"]]
target = target.to_numpy()
print(target)
consul = target[:, 0]
auto = target[:, 1]
power = target[:, 2]
print(features)
```

```
             dates  Factor A  Factor D    Factor F  Factor G  Factor H  \
0       1700-01-01    502.52     947.6     79050.0    502.10    502.73
1       1700-01-02    503.33     928.6     31082.0    502.28    501.96
2       1700-01-03    500.62     935.5     19375.0    502.09    499.17
3       1700-01-04    502.08     923.5     22010.0    501.88    500.43
4       1700-01-05    502.81     918.1     26533.0    501.70    501.46
…              …         …         …           …         …         …
161763  2142-11-23  10499.99  100499.9  10000500.0    599.99   1499.99
161764  2142-11-24  10499.99  100499.9  10000500.0    599.99   1499.99
161765  2142-11-25  10499.99  100499.9  10000500.0    599.99   1499.99
161766  2142-11-26  10499.99  100499.9  10000500.0    599.99   1499.99
161767  2142-11-27  10499.99  100499.9  10000500.0    599.99   1499.99

        TGD Consultancy Share price  TGD Automobiles Share price  \
0                             519.0                         420.0
1                             518.0                         420.0
2                             523.0                         437.0
3                             522.0                         437.0
4                             522.0                         437.0
…                             …                             …
161763                        498.0                         420.0
161764                        502.0                         420.0
161765                        508.0                         420.0
161766                        507.0                         420.0
161767                        499.0                         420.0

        TGD Power Share price
0                       507.0
1                       507.0
2                       522.0
3                       522.0
4                       522.0
…                       …
161763                  507.0
161764                  507.0
161765                  507.0
161766                  507.0
161767                  507.0

[161768 rows x 9 columns]
```

```
[[1.049999e+04 1.004999e+05 1.000050e+07 5.999900e+02 1.499990e+03]
 [1.049999e+04 1.004999e+05 1.000050e+07 5.999900e+02 1.499990e+03]
 [1.049999e+04 1.004999e+05 1.000050e+07 5.999900e+02 1.499990e+03]
 …
 [5.018400e+02 9.039000e+02 1.939730e+05 5.012700e+02 4.974600e+02]
 [4.964000e+02 9.290000e+02 1.675640e+05 5.011600e+02 4.995700e+02]
 [4.969400e+02 8.957000e+02 1.599330e+05 5.012500e+02 5.010300e+02]]
```

```
[[519. 420. 507.]
 [518. 420. 507.]
 [523. 437. 522.]
 …
 [508. 420. 507.]
 [507. 420. 507.]
 [499. 420. 507.]]
[[5.025200e+02 9.476000e+02 7.905000e+04 5.021000e+02 5.027300e+02]
 [5.033300e+02 9.286000e+02 3.108200e+04 5.022800e+02 5.019600e+02]
 [5.006200e+02 9.355000e+02 1.937500e+04 5.020900e+02 4.991700e+02]
 …
 [1.049999e+04 1.004999e+05 1.000050e+07 5.999900e+02 1.499990e+03]
 [1.049999e+04 1.004999e+05 1.000050e+07 5.999900e+02 1.499990e+03]
 [1.049999e+04 1.004999e+05 1.000050e+07 5.999900e+02 1.499990e+03]]
```

[ ]:
```
logModel = LogisticRegression(solver = "lbfgs", max_iter = 100)
logModel.fit(features[:-1000], consul[:-1000])
y_model = logModel.predict(features[-1000:])
print(mean_squared_error(y_model, consul[-1000:])**0.5)
```

```
8.696781013685467

/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[ ]:
```
logModel = LogisticRegression(solver = "lbfgs", max_iter = 100)
logModel.fit(features, power)
y_model = logModel.predict(test)
print(y_model)
```

```
[503. 503. 503. … 503. 503. 503.]

/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
```

```
  n_iter_i = _check_optimize_result(
```

[ ]:
```
c = y_model[y_model == 503]
len(c)
```

[ ]:
```
30000
```

[ ]:
```
logModel.fit(features, auto)
y_model1 = logModel.predict(test)
print(y_model1)
```

```
[407. 407. 407. … 407. 407. 407.]

/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[ ]:
```
c = y_model1[y_model1 == 407]
len(c)
```

[ ]:
```
30000
```

[ ]:
```
c = y_model[y_model == 508]
len(c)
```

[ ]:
```
26367
```

[ ]:
```
logModel = LogisticRegression(solver = "lbfgs", max_iter = 300)
logModel.fit(features, power)
y_model1 = logModel.predict(test)
print(y_model1)
```

```
[503. 503. 503. … 503. 503. 503.]

/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
```

```
regression
  n_iter_i = _check_optimize_result(
```

[ ]:
```
logModel.fit(features, auto)
y_model2 = logModel.predict(test)
print(y_model2)
```

```
[407. 407. 407. … 407. 407. 407.]

/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[ ]:
```
logModel.fit(features, consul)
y_model3 = logModel.predict(test)
print(y_model3)
```

```
[507. 507. 507. … 507. 507. 507.]

/usr/local/lib/python3.9/dist-packages/sklearn/linear_model/_logistic.py:458:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  n_iter_i = _check_optimize_result(
```

[ ]:
```
xmodel = xgb.XGBRegressor()
xmodel.fit(features, auto)
y_model2 = xmodel.predict(test)
print(y_model2)
```

```
[417.98288 417.98288 417.98288 … 420.5201  416.39923 416.4249 ]
```

[ ]:
```
import xgboost as xgb
import optuna
import warnings
warnings.filterwarnings('ignore')
```

[ ]:
```
def objective(trial):

    params = {
        'max_depth': trial.suggest_int('max_depth', 1, 9),
        'learning_rate': trial.suggest_loguniform('learning_rate', 0.01, 1.0),
        'n_estimators': trial.suggest_int('n_estimators', 50, 500),
        'min_child_weight': trial.suggest_int('min_child_weight', 1, 10),
        'gamma': trial.suggest_loguniform('gamma', 1e-8, 1.0),
        'subsample': trial.suggest_loguniform('subsample', 0.01, 1.0),
        'colsample_bytree': trial.suggest_loguniform('colsample_bytree', 0.01,␣
↪1.0),
        'reg_alpha': trial.suggest_loguniform('reg_alpha', 1e-8, 1.0),
        'reg_lambda': trial.suggest_loguniform('reg_lambda', 1e-8, 1.0),
        'eval_metric': 'mlogloss',
        'use_label_encoder': False
    }

    # Fit the model
    optuna_model = xgb.XGBRegressor(**params)
    optuna_model.fit(features[:-1000], auto[:-1000])

    # Make predictions
    y_pred = optuna_model.predict(features[-1000:])

    # Evaluate predictions
    accuracy = mean_squared_error(auto[-1000:], y_pred)**0.5
    return accuracy
```

[ ]:
```
study = optuna.create_study(direction = "minimize")
```

```
[I 2023-04-16 02:42:11,976] A new study created in memory with name:
no-name-2b8ff384-c8ef-4406-a0d8-59a78270bdd7
```

[ ]:
```
study.optimize(objective, n_trials=100)
```

```
[I 2023-04-16 02:42:18,438] Trial 0 finished with value:
11.267582439844356 and parameters: {'max_depth': 3, 'learning_rate':
0.013458669638579304, 'n_estimators': 273, 'min_child_weight': 5, 'gamma':
0.00377671183131739, 'subsample': 0.03741394450334979, 'colsample_bytree':
0.33726036929735553, 'reg_alpha': 4.668739711897681e-05, 'reg_lambda':
1.5238507832661963e-07}. Best is trial 0 with value: 11.267582439844356.
[I 2023-04-16 02:42:20,937] Trial 1 finished with value:
11.319138328465735 and parameters: {'max_depth': 5, 'learning_rate':
0.9039185492900466, 'n_estimators': 101, 'min_child_weight': 6, 'gamma':
0.01262842743045886, 'subsample': 0.16717757946024772, 'colsample_bytree':
0.030681401256252547, 'reg_alpha': 0.0005214474749087132, 'reg_lambda':
3.975047593466817e-08}. Best is trial 0 with value: 11.267582439844356.
```

```
[I 2023-04-16 02:42:26,723] Trial 2 finished with value:
10.776556296233402 and parameters: {'max_depth': 1, 'learning_rate':
0.014762076425806342, 'n_estimators': 466, 'min_child_weight': 5, 'gamma':
0.06542299621757734, 'subsample': 0.22313796999665944, 'colsample_bytree':
0.1612015195984227, 'reg_alpha': 2.8975109244753173e-08, 'reg_lambda':
5.978297030597595e-06}. Best is trial 2 with value: 10.776556296233402.
[I 2023-04-16 02:42:35,838] Trial 3 finished with value:
14.008142004950317 and parameters: {'max_depth': 4, 'learning_rate':
0.5996816150997832, 'n_estimators': 471, 'min_child_weight': 3, 'gamma':
0.07028943403228609, 'subsample': 0.026085903479626152, 'colsample_bytree':
0.01024826494846613, 'reg_alpha': 0.7187326094787205, 'reg_lambda':
5.3852097319945334e-08}. Best is trial 2 with value: 10.776556296233402.
[I 2023-04-16 02:42:44,917] Trial 4 finished with value:
10.974516866531511 and parameters: {'max_depth': 7, 'learning_rate':
0.14745454990007184, 'n_estimators': 226, 'min_child_weight': 9, 'gamma':
0.8739798301688814, 'subsample': 0.27648950711683234, 'colsample_bytree':
0.3228749084493528, 'reg_alpha': 1.1908893334503582e-07, 'reg_lambda':
1.4385633702157937e-06}. Best is trial 2 with value: 10.776556296233402.
[I 2023-04-16 02:42:52,550] Trial 5 finished with value:
10.910670509920287 and parameters: {'max_depth': 7, 'learning_rate':
0.05638386973638399, 'n_estimators': 226, 'min_child_weight': 7, 'gamma':
0.6363561699117684, 'subsample': 0.08293362671201464, 'colsample_bytree':
0.03367139342015467, 'reg_alpha': 0.368737503376115, 'reg_lambda':
0.06337745327914981}. Best is trial 2 with value: 10.776556296233402.
[I 2023-04-16 02:42:57,540] Trial 6 finished with value:
21.728220801899507 and parameters: {'max_depth': 8, 'learning_rate':
0.014880922035837893, 'n_estimators': 190, 'min_child_weight': 1, 'gamma':
0.0610903267769828, 'subsample': 0.14636242702716362, 'colsample_bytree':
0.10270355466650508, 'reg_alpha': 1.747025319924446e-06, 'reg_lambda':
0.0691080115306936l}. Best is trial 2 with value: 10.776556296233402.
[I 2023-04-16 02:43:00,917] Trial 7 finished with value:
10.941416187804963 and parameters: {'max_depth': 1, 'learning_rate':
0.055351452525236426, 'n_estimators': 234, 'min_child_weight': 2, 'gamma':
0.0001483593606164237, 'subsample': 0.09712661421777959, 'colsample_bytree':
0.3398015104503334, 'reg_alpha': 1.0025516206627096e-07, 'reg_lambda':
0.019243470163073227}. Best is trial 2 with value: 10.776556296233402.
[I 2023-04-16 02:43:05,680] Trial 8 finished with value:
11.015247856803954 and parameters: {'max_depth': 4, 'learning_rate':
0.053200927684149565, 'n_estimators': 265, 'min_child_weight': 9, 'gamma':
2.9080921707137084e-08, 'subsample': 0.012733989982585786, 'colsample_bytree':
0.02099771439548423, 'reg_alpha': 2.1631449004170006e-05, 'reg_lambda':
0.004443338145110827}. Best is trial 2 with value: 10.776556296233402.
[I 2023-04-16 02:43:13,812] Trial 9 finished with value:
12.217387207686999 and parameters: {'max_depth': 4, 'learning_rate':
0.44624341171896986, 'n_estimators': 380, 'min_child_weight': 7, 'gamma':
1.8721591933327447e-05, 'subsample': 0.02688315001428489, 'colsample_bytree':
0.018248245592090873, 'reg_alpha': 0.0010542280186736465, 'reg_lambda':
0.00033312349967141124}. Best is trial 2 with value: 10.776556296233402.
```

```
[I 2023-04-16 02:43:19,842] Trial 10 finished with value:
10.096791777828477 and parameters: {'max_depth': 1, 'learning_rate':
0.010950906154830574, 'n_estimators': 466, 'min_child_weight': 4, 'gamma':
0.00051145450126074, 'subsample': 0.7812610631368658, 'colsample_bytree':
0.7002947973180673, 'reg_alpha': 1.4507942241074427e-08, 'reg_lambda':
1.0789871790437864e-05}. Best is trial 10 with value: 10.096791777828477.
[I 2023-04-16 02:43:27,718] Trial 11 finished with value:
10.04107746919662 and parameters: {'max_depth': 1, 'learning_rate':
0.010068992498700352, 'n_estimators': 498, 'min_child_weight': 4, 'gamma':
0.0006242022030496838, 'subsample': 0.9021383968675993, 'colsample_bytree':
0.9211357400675775, 'reg_alpha': 1.8603550913841107e-08, 'reg_lambda':
1.0141077307660463e-05}. Best is trial 11 with value: 10.04107746919662.
[I 2023-04-16 02:43:37,780] Trial 12 finished with value:
9.925311401953982 and parameters: {'max_depth': 2, 'learning_rate':
0.010390325233089035, 'n_estimators': 399, 'min_child_weight': 4, 'gamma':
0.000240529596323366, 'subsample': 0.8584180697269808, 'colsample_bytree':
0.9498522910213606, 'reg_alpha': 1.3488119053153405e-08, 'reg_lambda':
2.8751063675914787e-05}. Best is trial 12 with value: 9.925311401953982.
[I 2023-04-16 02:43:46,945] Trial 13 finished with value:
10.90118720755878 and parameters: {'max_depth': 2, 'learning_rate':
0.026948520517056145, 'n_estimators': 369, 'min_child_weight': 3, 'gamma':
2.2188946633234072e-05, 'subsample': 0.947811355343832, 'colsample_bytree':
0.9692700226470268, 'reg_alpha': 6.095080254956782e-07, 'reg_lambda':
0.00017719846103469175}. Best is trial 12 with value: 9.925311401953982.
[I 2023-04-16 02:43:54,489] Trial 14 finished with value:
10.900468394536391 and parameters: {'max_depth': 2, 'learning_rate':
0.023772600086456767, 'n_estimators': 383, 'min_child_weight': 4, 'gamma':
0.0005728373182739482, 'subsample': 0.4121329562004333, 'colsample_bytree':
0.6498139563073405, 'reg_alpha': 1.2966780393279388e-06, 'reg_lambda':
3.668696105411863e-05}. Best is trial 12 with value: 9.925311401953982.
[I 2023-04-16 02:44:05,851] Trial 15 finished with value:
9.802996525523547 and parameters: {'max_depth': 2, 'learning_rate':
0.010617136233429983, 'n_estimators': 426, 'min_child_weight': 1, 'gamma':
7.1051573587471736e-06, 'subsample': 0.5854477453820605, 'colsample_bytree':
0.8621135292957185, 'reg_alpha': 1.3016728953025433e-08, 'reg_lambda':
1.065077128245049e-06}. Best is trial 15 with value: 9.802996525523547.
[I 2023-04-16 02:44:15,125] Trial 16 finished with value:
10.891506535337266 and parameters: {'max_depth': 3, 'learning_rate':
0.024720038869396145, 'n_estimators': 333, 'min_child_weight': 1, 'gamma':
2.8577950125624067e-06, 'subsample': 0.44744022833261105, 'colsample_bytree':
0.47775610110086447, 'reg_alpha': 1.3577975322478598e-08, 'reg_lambda':
7.133536396400199e-07}. Best is trial 15 with value: 9.802996525523547.
[I 2023-04-16 02:44:45,116] Trial 17 finished with value:
10.857637351488636 and parameters: {'max_depth': 6, 'learning_rate':
0.024764782967254786, 'n_estimators': 408, 'min_child_weight': 2, 'gamma':
1.2177401631489274e-06, 'subsample': 0.4845229388996947, 'colsample_bytree':
0.9965267244279753, 'reg_alpha': 4.706417610776018e-07, 'reg_lambda':
3.4914048496803496e-07}. Best is trial 15 with value: 9.802996525523547.
```

```
[I 2023-04-16 02:45:02,938] Trial 18 finished with value:
14.168284101613565 and parameters: {'max_depth': 9, 'learning_rate':
0.01006215256102655, 'n_estimators': 329, 'min_child_weight': 2, 'gamma':
1.2430352048631412e-07, 'subsample': 0.6070146376439193, 'colsample_bytree':
0.19080867382441916, 'reg_alpha': 3.674154065622822e-06, 'reg_lambda':
2.0458828869141582e-08}. Best is trial 15 with value: 9.802996525523547.
[I 2023-04-16 02:45:13,545] Trial 19 finished with value:
10.958212577659461 and parameters: {'max_depth': 3, 'learning_rate':
0.11870241004976101, 'n_estimators': 432, 'min_child_weight': 7, 'gamma':
3.257037783698835e-05, 'subsample': 0.30780201470063023, 'colsample_bytree':
0.49356310985490087, 'reg_alpha': 1.3788651382930815e-07, 'reg_lambda':
0.0008775913282605641}. Best is trial 15 with value: 9.802996525523547.
[I 2023-04-16 02:45:19,945] Trial 20 finished with value:
10.937428326995848 and parameters: {'max_depth': 2, 'learning_rate':
0.03678485368905384, 'n_estimators': 320, 'min_child_weight': 10, 'gamma':
3.0740314430266777e-06, 'subsample': 0.5496905733221962, 'colsample_bytree':
0.06883279076737621, 'reg_alpha': 7.038994477623286e-06, 'reg_lambda':
0.711386115596718}. Best is trial 15 with value: 9.802996525523547.
[I 2023-04-16 02:45:32,058] Trial 21 finished with value:
10.834577061340536 and parameters: {'max_depth': 2, 'learning_rate':
0.015624739137224569, 'n_estimators': 491, 'min_child_weight': 4, 'gamma':
0.0001869016509161799, 'subsample': 0.7866736039007367, 'colsample_bytree':
0.984488725519543, 'reg_alpha': 1.3586440134445346e-08, 'reg_lambda':
3.45230352761672e-05}. Best is trial 15 with value: 9.802996525523547.
[I 2023-04-16 02:45:36,856] Trial 22 finished with value:
9.77140820699839 and parameters: {'max_depth': 1, 'learning_rate':
0.010325220156445191, 'n_estimators': 429, 'min_child_weight': 3, 'gamma':
0.0015327730384227147, 'subsample': 0.9040614267451695, 'colsample_bytree':
0.5923716765914521, 'reg_alpha': 5.6492636606637425e-08, 'reg_lambda':
2.4803611008645508e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:45:48,041] Trial 23 finished with value:
10.846830883406152 and parameters: {'max_depth': 3, 'learning_rate':
0.017910599285335525, 'n_estimators': 428, 'min_child_weight': 3, 'gamma':
0.002124675227271941, 'subsample': 0.9387522081830338, 'colsample_bytree':
0.5685421242655224, 'reg_alpha': 9.330553284853684e-08, 'reg_lambda':
2.2990199315667242e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:45:58,311] Trial 24 finished with value:
10.838438529389533 and parameters: {'max_depth': 2, 'learning_rate':
0.017648679282106255, 'n_estimators': 428, 'min_child_weight': 1, 'gamma':
9.016695407302865e-05, 'subsample': 0.5990370843944255, 'colsample_bytree':
0.6496139498595246, 'reg_alpha': 3.235122438228392e-07, 'reg_lambda':
3.0453916822032094e-07}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:12,611] Trial 25 finished with value:
10.678853676799841 and parameters: {'max_depth': 5, 'learning_rate':
0.010659642150642263, 'n_estimators': 359, 'min_child_weight': 2, 'gamma':
0.0021475800982883957, 'subsample': 0.3784927117823603, 'colsample_bytree':
0.42432380120733365, 'reg_alpha': 3.800587448117396e-08, 'reg_lambda':
2.3993248773705337e-06}. Best is trial 22 with value: 9.77140820699839.
```

```
[I 2023-04-16 02:46:13,877] Trial 26 finished with value:
49.07449980442877 and parameters: {'max_depth': 1, 'learning_rate':
0.019451834529528676, 'n_estimators': 105, 'min_child_weight': 3, 'gamma':
1.0311941905245797e-05, 'subsample': 0.6386078262985767, 'colsample_bytree':
0.25163765669140903, 'reg_alpha': 5.417059030146142e-08, 'reg_lambda':
4.362501353561873e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:26,126] Trial 27 finished with value:
10.91484265749668 and parameters: {'max_depth': 3, 'learning_rate':
0.03590389314230892, 'n_estimators': 402, 'min_child_weight': 6, 'gamma':
5.7723247078336695e-05, 'subsample': 0.9957263418810318, 'colsample_bytree':
0.7222810110957563, 'reg_alpha': 1.0408729849449076e-08, 'reg_lambda':
1.4814139827072813e-07}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:32,510] Trial 28 finished with value:
10.839201876955947 and parameters: {'max_depth': 2, 'learning_rate':
0.012364674819518525, 'n_estimators': 304, 'min_child_weight': 1, 'gamma':
6.583390318407521e-07, 'subsample': 0.35695742032304695, 'colsample_bytree':
0.4731894774020181, 'reg_alpha': 6.081960957122083e-07, 'reg_lambda':
1.0696514655930206e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:36,257] Trial 29 finished with value:
63.137265757153514 and parameters: {'max_depth': 4, 'learning_rate':
0.012521749812111873, 'n_estimators': 145, 'min_child_weight': 5, 'gamma':
6.414454233792415e-06, 'subsample': 0.6495626176518511, 'colsample_bytree':
0.33903620606215784, 'reg_alpha': 1.810618062889993e-07, 'reg_lambda':
6.5107014990118735e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:38,011] Trial 30 finished with value:
175.18832660461712 and parameters: {'max_depth': 3, 'learning_rate':
0.014178290356784187, 'n_estimators': 59, 'min_child_weight': 2, 'gamma':
5.6753539406749454e-05, 'subsample': 0.4488477255252794, 'colsample_bytree':
0.7269588519507878, 'reg_alpha': 4.9430267624247475e-08, 'reg_lambda':
1.6992244385293253e-07}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:46,173] Trial 31 finished with value:
10.203292522090951 and parameters: {'max_depth': 1, 'learning_rate':
0.010557507492743385, 'n_estimators': 500, 'min_child_weight': 4, 'gamma':
0.0005620622408036146, 'subsample': 0.7608967375308815, 'colsample_bytree':
0.828220312390952, 'reg_alpha': 4.518673392620937e-08, 'reg_lambda':
1.0552130830769551e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:52,781] Trial 32 finished with value:
9.808473826548001 and parameters: {'max_depth': 1, 'learning_rate':
0.010277286837992481, 'n_estimators': 446, 'min_child_weight': 5, 'gamma':
0.005328973093494086, 'subsample': 0.9533867297620254, 'colsample_bytree':
0.610846895117139, 'reg_alpha': 1.3986491452148314e-08, 'reg_lambda':
6.082065742022304e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:46:58,348] Trial 33 finished with value:
10.934513000368916 and parameters: {'max_depth': 1, 'learning_rate':
0.01891823214579977, 'n_estimators': 449, 'min_child_weight': 5, 'gamma':
0.011382866989787962, 'subsample': 0.54188797635399, 'colsample_bytree':
0.5937389332593662, 'reg_alpha': 1.027070099226026e-08, 'reg_lambda':
6.999763093404266e-05}. Best is trial 22 with value: 9.77140820699839.
```

```
[I 2023-04-16 02:47:05,790] Trial 34 finished with value:
10.332407928855075 and parameters: {'max_depth': 2, 'learning_rate':
0.013678113703320139, 'n_estimators': 405, 'min_child_weight': 6, 'gamma':
0.005068684931872217, 'subsample': 0.7245343893474376, 'colsample_bytree':
0.3756572219715716, 'reg_alpha': 2.2764273069905936e-07, 'reg_lambda':
4.550988986491082e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:47:10,454] Trial 35 finished with value:
10.701319564130644 and parameters: {'max_depth': 1, 'learning_rate':
0.014312915928350411, 'n_estimators': 452, 'min_child_weight': 5, 'gamma':
0.00021115690531475965, 'subsample': 0.2310999833509517, 'colsample_bytree':
0.5192286274741783, 'reg_alpha': 4.4457984772960295e-08, 'reg_lambda':
7.211419956216215e-07}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:47:16,907] Trial 36 finished with value:
10.677428844017678 and parameters: {'max_depth': 2, 'learning_rate':
0.01848690507179873, 'n_estimators': 349, 'min_child_weight': 6, 'gamma':
0.006983217839876853, 'subsample': 0.32339461858902174, 'colsample_bytree':
0.2790513545879806, 'reg_alpha': 3.1586020899389505e-08, 'reg_lambda':
2.38710815011893e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:47:33,060] Trial 37 finished with value:
10.161340261552674 and parameters: {'max_depth': 5, 'learning_rate':
0.01324975558150716, 'n_estimators': 395, 'min_child_weight': 3, 'gamma':
0.0025843389496359743, 'subsample': 0.7225760949189899, 'colsample_bytree':
0.4072439229325148, 'reg_alpha': 9.909749280278268e-08, 'reg_lambda':
6.44175442335824e-08}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:47:37,698] Trial 38 finished with value:
9.958513304304507 and parameters: {'max_depth': 1, 'learning_rate':
0.016347293867310972, 'n_estimators': 298, 'min_child_weight': 3, 'gamma':
0.03278857486268327, 'subsample': 0.2500316507506338, 'colsample_bytree':
0.7917055756638152, 'reg_alpha': 3.317495118963916e-08, 'reg_lambda':
0.00011173973079419741}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:47:46,512] Trial 39 finished with value:
10.932134587612625 and parameters: {'max_depth': 3, 'learning_rate':
0.03242205967138578, 'n_estimators': 469, 'min_child_weight': 5, 'gamma':
0.22981912139190927, 'subsample': 0.16805623324637003, 'colsample_bytree':
0.5816207794820216, 'reg_alpha': 2.945172194815018e-07, 'reg_lambda':
3.5922328446201097e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:47:52,846] Trial 40 finished with value:
10.94200939361157 and parameters: {'max_depth': 1, 'learning_rate':
0.020961615031875037, 'n_estimators': 443, 'min_child_weight': 8, 'gamma':
0.027140931501006146, 'subsample': 0.5047198108131667, 'colsample_bytree':
0.26899759663417383, 'reg_alpha': 9.743163868669916e-08, 'reg_lambda':
1.6092664625218557e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:47:56,005] Trial 41 finished with value:
9.823890098268471 and parameters: {'max_depth': 1, 'learning_rate':
0.015615249442510656, 'n_estimators': 269, 'min_child_weight': 4, 'gamma':
0.026361405661311912, 'subsample': 0.2724273323973324, 'colsample_bytree':
0.7696425045117944, 'reg_alpha': 2.7452498212042476e-08, 'reg_lambda':
0.0001098218973588791}. Best is trial 22 with value: 9.77140820699839.
```

```
[I 2023-04-16 02:48:02,677] Trial 42 finished with value:
11.520697282255489 and parameters: {'max_depth': 2, 'learning_rate':
0.012920115215505712, 'n_estimators': 280, 'min_child_weight': 4, 'gamma':
0.014736508935406265, 'subsample': 0.3905128648136873, 'colsample_bytree':
0.7725934482396636, 'reg_alpha': 1.0108814802725418e-08, 'reg_lambda':
0.00029277329684873767}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:48:06,085] Trial 43 finished with value:
20.006685625758795 and parameters: {'max_depth': 1, 'learning_rate':
0.011852379899052468, 'n_estimators': 245, 'min_child_weight': 5, 'gamma':
0.11866961584422618, 'subsample': 0.8006576551138826, 'colsample_bytree':
0.8003032612015979, 'reg_alpha': 2.6858553578211195e-08, 'reg_lambda':
8.973445484161033e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:48:16,141] Trial 44 finished with value:
9.845408983236718 and parameters: {'max_depth': 2, 'learning_rate':
0.010164226165873562, 'n_estimators': 419, 'min_child_weight': 4, 'gamma':
0.0035863950594918785, 'subsample': 0.9541464484306094, 'colsample_bytree':
0.578262519474316, 'reg_alpha': 8.567706774836794e-08, 'reg_lambda':
2.07166737563972e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:48:18,364] Trial 45 finished with value:
26.59703393182823 and parameters: {'max_depth': 1, 'learning_rate':
0.01523801030025432, 'n_estimators': 172, 'min_child_weight': 6, 'gamma':
0.0014618699731402567, 'subsample': 0.5699952764943645, 'colsample_bytree':
0.40937211422710784, 'reg_alpha': 1.7011840944321416e-07, 'reg_lambda':
3.7278952975323586e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:48:28,173] Trial 46 finished with value:
9.99355338901093 and parameters: {'max_depth': 6, 'learning_rate':
0.021546711946558272, 'n_estimators': 192, 'min_child_weight': 4, 'gamma':
0.004432497551339036, 'subsample': 0.9525120399018527, 'colsample_bytree':
0.5976682299016087, 'reg_alpha': 9.04933410335513e-07, 'reg_lambda':
1.09154787655588e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:48:36,402] Trial 47 finished with value:
26.647874786918585 and parameters: {'max_depth': 4, 'learning_rate':
0.010251810010999552, 'n_estimators': 257, 'min_child_weight': 3, 'gamma':
0.010521615857487617, 'subsample': 0.2731490263758657, 'colsample_bytree':
0.5114290925278077, 'reg_alpha': 8.031773492340618e-08, 'reg_lambda':
0.0006340862449603307}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:48:41,637] Trial 48 finished with value:
10.851136320879458 and parameters: {'max_depth': 1, 'learning_rate':
0.015540581351334397, 'n_estimators': 475, 'min_child_weight': 2, 'gamma':
0.00098916691766602, 'subsample': 0.6777755981915885, 'colsample_bytree':
0.32068769214413995, 'reg_alpha': 3.352160645255874e-07, 'reg_lambda':
1.214033838894877e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:06,316] Trial 49 finished with value:
10.884003890107142 and parameters: {'max_depth': 6, 'learning_rate':
0.027732991031457876, 'n_estimators': 417, 'min_child_weight': 5, 'gamma':
0.0038672215588778087, 'subsample': 0.4492627255474678, 'colsample_bytree':
0.6773839742697342, 'reg_alpha': 2.278594058006403e-06, 'reg_lambda':
2.025689469123804e-05}. Best is trial 22 with value: 9.77140820699839.
```

```
[I 2023-04-16 02:49:11,339] Trial 50 finished with value:
30.529028715828694 and parameters: {'max_depth': 2, 'learning_rate':
0.012010385015035058, 'n_estimators': 208, 'min_child_weight': 7, 'gamma':
0.02101333412386255, 'subsample': 0.19379376088066336, 'colsample_bytree':
0.8747443193900447, 'reg_alpha': 2.3318942122188658e-08, 'reg_lambda':
6.017030390648666e-06}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:19,462] Trial 51 finished with value:
10.214558087155977 and parameters: {'max_depth': 2, 'learning_rate':
0.01048783084228974, 'n_estimators': 378, 'min_child_weight': 4, 'gamma':
0.001058689338847781, 'subsample': 0.9998550423230528, 'colsample_bytree':
0.8850670335113764, 'reg_alpha': 2.819477640897811e-08, 'reg_lambda':
5.803384009924751e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:25,453] Trial 52 finished with value:
9.781471837477666 and parameters: {'max_depth': 1, 'learning_rate':
0.012828664205057703, 'n_estimators': 350, 'min_child_weight': 4, 'gamma':
0.0003172838524638698, 'subsample': 0.8332367294792443, 'colsample_bytree':
0.9783205325323455, 'reg_alpha': 2.1117439464548888e-08, 'reg_lambda':
0.00015247537700350972}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:29,071] Trial 53 finished with value:
9.8184687566912 and parameters: {'max_depth': 1, 'learning_rate':
0.01623090307963363, 'n_estimators': 283, 'min_child_weight': 4, 'gamma':
0.0004057938028776573, 'subsample': 0.8008427977522136, 'colsample_bytree':
0.6539832277928248, 'reg_alpha': 7.104617394288536e-08, 'reg_lambda':
0.00015824496980450847}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:34,078] Trial 54 finished with value:
10.691394921600065 and parameters: {'max_depth': 1, 'learning_rate':
0.0220944600754105, 'n_estimators': 289, 'min_child_weight': 3, 'gamma':
0.000321075912782568, 'subsample': 0.7196166767921155, 'colsample_bytree':
0.7004458371777487, 'reg_alpha': 1.6961686083688102e-08, 'reg_lambda':
0.00026449887421570856}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:38,286] Trial 55 finished with value:
10.37054822756396 and parameters: {'max_depth': 1, 'learning_rate':
0.016449653033069927, 'n_estimators': 338, 'min_child_weight': 3, 'gamma':
0.00011731166568318582, 'subsample': 0.5288053525337939, 'colsample_bytree':
0.4911861121386479, 'reg_alpha': 2.107449222102685e-08, 'reg_lambda':
0.00015305076264217191}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:42,735] Trial 56 finished with value:
9.816320349451605 and parameters: {'max_depth': 1, 'learning_rate':
0.01369933621853289, 'n_estimators': 308, 'min_child_weight': 1, 'gamma':
0.0004097740037364815, 'subsample': 0.622747824797819, 'colsample_bytree':
0.9833956888468103, 'reg_alpha': 2.0027020454918446e-07, 'reg_lambda':
0.00010343146649806103}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:49:48,224] Trial 57 finished with value:
10.244917757594399 and parameters: {'max_depth': 1, 'learning_rate':
0.01263355206266082, 'n_estimators': 310, 'min_child_weight': 1, 'gamma':
0.0004789323798809778, 'subsample': 0.8055537648913281, 'colsample_bytree':
0.9730057791198529, 'reg_alpha': 1.687573190671078e-07, 'reg_lambda':
0.0011672626644088282}. Best is trial 22 with value: 9.77140820699839.
```

```
[I 2023-04-16 02:50:21,429] Trial 58 finished with value:
10.692759413624625 and parameters: {'max_depth': 9, 'learning_rate':
0.01940532764954864, 'n_estimators': 342, 'min_child_weight': 1, 'gamma':
0.00030516439790317945, 'subsample': 0.6351935732931443, 'colsample_bytree':
0.6521586277308763, 'reg_alpha': 6.546164732539858e-07, 'reg_lambda':
0.0005289400949383806}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:50:26,281] Trial 59 finished with value:
10.948474989275123 and parameters: {'max_depth': 1, 'learning_rate':
0.026739610190498277, 'n_estimators': 358, 'min_child_weight': 2, 'gamma':
0.0010714718033620727, 'subsample': 0.8419512131396732, 'colsample_bytree':
0.8227919708806185, 'reg_alpha': 7.326702763152607e-08, 'reg_lambda':
5.067508694056319e-05}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:50:36,188] Trial 60 finished with value:
9.830647930346709 and parameters: {'max_depth': 2, 'learning_rate':
0.012015079460795957, 'n_estimators': 385, 'min_child_weight': 1, 'gamma':
0.00010020883063366901, 'subsample': 0.46118418541591577, 'colsample_bytree':
0.9830252155782303, 'reg_alpha': 3.171206501527268e-07, 'reg_lambda':
0.00023603917680763487}. Best is trial 22 with value: 9.77140820699839.
[I 2023-04-16 02:50:39,931] Trial 61 finished with value:
9.769342473120386 and parameters: {'max_depth': 1, 'learning_rate':
0.016116763317901944, 'n_estimators': 273, 'min_child_weight': 2, 'gamma':
0.0007411344904969246, 'subsample': 0.5670769278693609, 'colsample_bytree':
0.7288416268210294, 'reg_alpha': 2.1751933888208096e-08, 'reg_lambda':
0.0001076801059578796}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:50:45,340] Trial 62 finished with value:
9.78235115381206 and parameters: {'max_depth': 1, 'learning_rate':
0.014217884374034926, 'n_estimators': 315, 'min_child_weight': 2, 'gamma':
0.00034553665430270535, 'subsample': 0.6059915953813432, 'colsample_bytree':
0.6516242552524115, 'reg_alpha': 5.526898533799315e-08, 'reg_lambda':
0.0019935432454171466}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:50:49,982] Trial 63 finished with value:
9.771070730785127 and parameters: {'max_depth': 1, 'learning_rate':
0.013714602052896149, 'n_estimators': 319, 'min_child_weight': 2, 'gamma':
0.00017036628842814782, 'subsample': 0.5904533608901273, 'colsample_bytree':
0.9964516307075565, 'reg_alpha': 1.848040139075623e-08, 'reg_lambda':
0.0012592984948594248}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:50:57,824] Trial 64 finished with value:
11.426746011984449 and parameters: {'max_depth': 2, 'learning_rate':
0.010005533596520262, 'n_estimators': 364, 'min_child_weight': 2, 'gamma':
0.00016065419036211427, 'subsample': 0.5210688776152774, 'colsample_bytree':
0.4438337010982044, 'reg_alpha': 1.4730774241930936e-08, 'reg_lambda':
0.001863770549268333}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:51:20,212] Trial 65 finished with value:
10.558949873784345 and parameters: {'max_depth': 8, 'learning_rate':
0.01183371387665859, 'n_estimators': 327, 'min_child_weight': 2, 'gamma':
3.4186587646466516e-05, 'subsample': 0.5935350653698234, 'colsample_bytree':
0.5398108001765234, 'reg_alpha': 5.382353027739368e-08, 'reg_lambda':
0.0014142266228145627}. Best is trial 61 with value: 9.769342473120386.
```

[I 2023-04-16 02:51:23,863] Trial 66 finished with value:
9.87180704274306 and parameters: {'max_depth': 1, 'learning_rate':
0.018293903134351353, 'n_estimators': 257, 'min_child_weight': 2, 'gamma':
0.0007560670461934805, 'subsample': 0.4052276916541737, 'colsample_bytree':
0.8394809337349076, 'reg_alpha': 1.871057406389554e-08, 'reg_lambda':
0.0059300120596671385}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:51:31,145] Trial 67 finished with value:
10.812785762953178 and parameters: {'max_depth': 2, 'learning_rate':
0.022878423154163968, 'n_estimators': 320, 'min_child_weight': 2, 'gamma':
0.0022612459029247613, 'subsample': 0.3399097187869839, 'colsample_bytree':
0.6955426763830295, 'reg_alpha': 4.4394614680262553e-08, 'reg_lambda':
0.0017362204175897153}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:51:39,686] Trial 68 finished with value:
10.774982421878224 and parameters: {'max_depth': 2, 'learning_rate':
0.014350950011104568, 'n_estimators': 487, 'min_child_weight': 3, 'gamma':
0.0015716969075750718, 'subsample': 0.8605255712449291, 'colsample_bytree':
0.4601999181962238, 'reg_alpha': 1.690777659104074e-08, 'reg_lambda':
0.00046698107491493937}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:51:43,502] Trial 69 finished with value:
25.043611773890326 and parameters: {'max_depth': 1, 'learning_rate':
0.011665673918206656, 'n_estimators': 230, 'min_child_weight': 2, 'gamma':
0.00023340590139719113, 'subsample': 0.680077513678318, 'colsample_bytree':
0.5659981501607232, 'reg_alpha': 1.220548332942551e-07, 'reg_lambda':
0.0035014722472598648}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:51:49,928] Trial 70 finished with value:
10.902711306763168 and parameters: {'max_depth': 1, 'learning_rate':
0.01775312795900371, 'n_estimators': 445, 'min_child_weight': 1, 'gamma':
0.000710543495712658, 'subsample': 0.4549768402990821, 'colsample_bytree':
0.8687591469583272, 'reg_alpha': 1.0962479809927062e-08, 'reg_lambda':
0.00037597260152596085}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:51:55,639] Trial 71 finished with value:
9.815311178991045 and parameters: {'max_depth': 1, 'learning_rate':
0.013786658537183116, 'n_estimators': 306, 'min_child_weight': 1, 'gamma':
6.286225375596269e-05, 'subsample': 0.6011571304622553, 'colsample_bytree':
0.9878906444888482, 'reg_alpha': 4.659103334568675e-08, 'reg_lambda':
7.153429509519303e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:51:59,748] Trial 72 finished with value:
9.79097506878862 and parameters: {'max_depth': 1, 'learning_rate':
0.014275853782159599, 'n_estimators': 299, 'min_child_weight': 1, 'gamma':
6.959657042541342e-05, 'subsample': 0.5566440599389002, 'colsample_bytree':
0.744674092647421, 'reg_alpha': 4.1026987317978544e-08, 'reg_lambda':
3.884565883312896e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:52:04,863] Trial 73 finished with value:
13.4791424005744 and parameters: {'max_depth': 1, 'learning_rate':
0.011405023015418979, 'n_estimators': 294, 'min_child_weight': 1, 'gamma':
0.00013528747059553806, 'subsample': 0.7262287769378248, 'colsample_bytree':
0.7357391690515225, 'reg_alpha': 2.6557610697681415e-08, 'reg_lambda':
3.9985196764331415e-05}. Best is trial 61 with value: 9.769342473120386.

[I 2023-04-16 02:52:14,463] Trial 74 finished with value:
10.710881136260959 and parameters: {'max_depth': 2, 'learning_rate':
0.01466061082834505, 'n_estimators': 455, 'min_child_weight': 2, 'gamma':
2.861849402481373e-05, 'subsample': 0.5124788085914532, 'colsample_bytree':
0.6436170400509563, 'reg_alpha': 1.1693483840224222e-07, 'reg_lambda':
0.0008264845575454558}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:52:18,994] Trial 75 finished with value:
10.451818233936809 and parameters: {'max_depth': 1, 'learning_rate':
0.02100667848684358, 'n_estimators': 272, 'min_child_weight': 3, 'gamma':
0.00022608863222176707, 'subsample': 0.8626734379724166, 'colsample_bytree':
0.7499708042706555, 'reg_alpha': 1.0267748849594354e-08, 'reg_lambda':
3.2056812085967475e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:52:25,679] Trial 76 finished with value:
10.69203566978195 and parameters: {'max_depth': 2, 'learning_rate':
0.016822651739447754, 'n_estimators': 390, 'min_child_weight': 1, 'gamma':
0.0006607820318290408, 'subsample': 0.3702930795880643, 'colsample_bytree':
0.504499894019111, 'reg_alpha': 5.8351747210013224e-08, 'reg_lambda':
1.8742729901719503e-06}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:52:31,134] Trial 77 finished with value:
9.86593341473659 and parameters: {'max_depth': 1, 'learning_rate':
0.013134572648761325, 'n_estimators': 316, 'min_child_weight': 2, 'gamma':
7.398006770735625e-05, 'subsample': 0.6735966947328559, 'colsample_bytree':
0.6054220742427444, 'reg_alpha': 3.164016297789487e-08, 'reg_lambda':
0.00019249335266618544}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:52:36,479] Trial 78 finished with value:
9.777269784770901 and parameters: {'max_depth': 1, 'learning_rate':
0.011606798816830619, 'n_estimators': 373, 'min_child_weight': 2, 'gamma':
4.804593131865883e-05, 'subsample': 0.4179350106988198, 'colsample_bytree':
0.8592384006339454, 'reg_alpha': 1.8117748715881763e-08, 'reg_lambda':
0.0004313832818004672}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:52:51,549] Trial 79 finished with value:
10.1174802724919 and parameters: {'max_depth': 2, 'learning_rate':
0.011421757696067533, 'n_estimators': 351, 'min_child_weight': 2, 'gamma':
1.9687587836777664e-05, 'subsample': 0.4215785507229551, 'colsample_bytree':
0.8670403798307268, 'reg_alpha': 2.052268400962511e-08, 'reg_lambda':
0.00047470261790389127}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:02,872] Trial 80 finished with value:
10.73412878121988 and parameters: {'max_depth': 3, 'learning_rate':
0.019536369081044416, 'n_estimators': 345, 'min_child_weight': 1, 'gamma':
4.4046041847381685e-05, 'subsample': 0.5692126489397046, 'colsample_bytree':
0.7373726809888063, 'reg_alpha': 1.5556539939735175e-07, 'reg_lambda':
0.0003130687800708415}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:09,241] Trial 81 finished with value:
10.426926436291 and parameters: {'max_depth': 1, 'learning_rate':
0.013223371198658247, 'n_estimators': 430, 'min_child_weight': 3, 'gamma':
1.3721209670094706e-05, 'subsample': 0.4871886183180671, 'colsample_bytree':
0.8704344698824809, 'reg_alpha': 3.9183014617006756e-08, 'reg_lambda':
0.00018272772148921436}. Best is trial 61 with value: 9.769342473120386.

[I 2023-04-16 02:53:14,882] Trial 82 finished with value:
9.852077533112322 and parameters: {'max_depth': 1, 'learning_rate':
0.011174485290131162, 'n_estimators': 373, 'min_child_weight': 2, 'gamma':
4.043620776044511e-05, 'subsample': 0.7856062379537618, 'colsample_bytree':
0.5505488203969833, 'reg_alpha': 1.611527099082859e-08, 'reg_lambda':
0.0007198427152023388}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:20,643] Trial 83 finished with value:
10.697346404841305 and parameters: {'max_depth': 1, 'learning_rate':
0.01468504606391823, 'n_estimators': 437, 'min_child_weight': 3, 'gamma':
8.457038309049402e-05, 'subsample': 0.7247654725595932, 'colsample_bytree':
0.6540277945085441, 'reg_alpha': 6.328187976647454e-08, 'reg_lambda':
7.540749739655845e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:26,047] Trial 84 finished with value:
10.349200962517575 and parameters: {'max_depth': 1, 'learning_rate':
0.01672662988913581, 'n_estimators': 330, 'min_child_weight': 2, 'gamma':
0.00017381552676637695, 'subsample': 0.9119709065407697, 'colsample_bytree':
0.7816588379289233, 'reg_alpha': 3.4349092407123754e-08, 'reg_lambda':
2.982136496829845e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:32,054] Trial 85 finished with value:
9.963601500253187 and parameters: {'max_depth': 1, 'learning_rate':
0.010129087839082262, 'n_estimators': 402, 'min_child_weight': 10, 'gamma':
0.00034733835641027904, 'subsample': 0.555977219642962, 'colsample_bytree':
0.8954843900431804, 'reg_alpha': 1.9322699337141603e-08, 'reg_lambda':
7.392845377664768e-06}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:40,151] Trial 86 finished with value:
10.256013453939126 and parameters: {'max_depth': 2, 'learning_rate':
0.012884248888344949, 'n_estimators': 418, 'min_child_weight': 1, 'gamma':
0.00011943962625711961, 'subsample': 0.30172467997664987, 'colsample_bytree':
0.400607215416404, 'reg_alpha': 6.418833610341141e-08, 'reg_lambda':
1.3042081386624555e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:46,283] Trial 87 finished with value:
10.817981171490429 and parameters: {'max_depth': 1, 'learning_rate':
0.01526418088300544, 'n_estimators': 460, 'min_child_weight': 3, 'gamma':
0.0015327558364597298, 'subsample': 0.6363911034953883, 'colsample_bytree':
0.6177268912988022, 'reg_alpha': 1.2118893630673253e-08, 'reg_lambda':
2.2161111551753213e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:50,629] Trial 88 finished with value:
22.777342725614744 and parameters: {'max_depth': 1, 'learning_rate':
0.011236932970152293, 'n_estimators': 247, 'min_child_weight': 1, 'gamma':
5.512602721085313e-05, 'subsample': 0.8880763870572164, 'colsample_bytree':
0.789348975246648, 'reg_alpha': 9.76495851440434e-08, 'reg_lambda':
2.8600712339525894e-06}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:53:55,564] Trial 89 finished with value:
10.170203219939953 and parameters: {'max_depth': 2, 'learning_rate':
0.017882737898249703, 'n_estimators': 292, 'min_child_weight': 2, 'gamma':
9.093359842303825e-06, 'subsample': 0.9955949669109234, 'colsample_bytree':
0.456777288234313, 'reg_alpha': 2.700442978542288e-08, 'reg_lambda':
4.940500403237696e-05}. Best is trial 61 with value: 9.769342473120386.

[I 2023-04-16 02:54:05,297] Trial 90 finished with value:
10.74636795883952 and parameters: {'max_depth': 2, 'learning_rate':
0.014191338664103827, 'n_estimators': 481, 'min_child_weight': 3, 'gamma':
2.1541025661734178e-05, 'subsample': 0.4140244768825499, 'colsample_bytree':
0.5201488295263255, 'reg_alpha': 3.949808141244162e-08, 'reg_lambda':
0.00011077244786251043}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:09,744] Trial 91 finished with value:
10.909822160606973 and parameters: {'max_depth': 1, 'learning_rate':
0.012316162592767074, 'n_estimators': 302, 'min_child_weight': 1, 'gamma':
5.3011548024272275e-05, 'subsample': 0.591248414120657, 'colsample_bytree':
0.9636268563569862, 'reg_alpha': 5.63613652226652e-08, 'reg_lambda':
6.368010059630199e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:15,037] Trial 92 finished with value:
10.599144022703666 and parameters: {'max_depth': 1, 'learning_rate':
0.013604476185001343, 'n_estimators': 279, 'min_child_weight': 1, 'gamma':
6.837297437320406e-05, 'subsample': 0.4927116724343896, 'colsample_bytree':
0.9994147917286167, 'reg_alpha': 1.9856000531631434e-08, 'reg_lambda':
0.00015222638881757283}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:19,196] Trial 93 finished with value:
11.731190352359175 and parameters: {'max_depth': 1, 'learning_rate':
0.011154768342599559, 'n_estimators': 320, 'min_child_weight': 1, 'gamma':
0.00017959170699222653, 'subsample': 0.7427979662856359, 'colsample_bytree':
0.7178064699506909, 'reg_alpha': 1.0258415293188113e-08, 'reg_lambda':
7.870014954035359e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:24,139] Trial 94 finished with value:
9.907949641774447 and parameters: {'max_depth': 1, 'learning_rate':
0.01573423642849437, 'n_estimators': 261, 'min_child_weight': 2, 'gamma':
9.722272764831533e-05, 'subsample': 0.6591804299862634, 'colsample_bytree':
0.8954237704452399, 'reg_alpha': 1.3107759749322154e-07, 'reg_lambda':
0.00030797325940883985}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:29,411] Trial 95 finished with value:
9.907401189222462 and parameters: {'max_depth': 1, 'learning_rate':
0.013396465510770235, 'n_estimators': 357, 'min_child_weight': 2, 'gamma':
0.0002589394798075473, 'subsample': 0.5723313277350773, 'colsample_bytree':
0.8028646389556406, 'reg_alpha': 4.341084876587288e-08, 'reg_lambda':
1.4762243878340769e-05}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:35,410] Trial 96 finished with value:
10.959622650257371 and parameters: {'max_depth': 1, 'learning_rate':
0.010035577283729754, 'n_estimators': 370, 'min_child_weight': 1, 'gamma':
0.00046804846057349524, 'subsample': 0.8133974476050608, 'colsample_bytree':
0.6838246838499444, 'reg_alpha': 2.5045849979924055e-08, 'reg_lambda':
8.507146690362741e-06}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:41,974] Trial 97 finished with value:
9.89447830863167 and parameters: {'max_depth': 2, 'learning_rate':
0.012345158086186669, 'n_estimators': 338, 'min_child_weight': 5, 'gamma':
3.15641040248306e-05, 'subsample': 0.36894049495720616, 'colsample_bytree':
0.6190647068622318, 'reg_alpha': 2.5469362727290516e-07, 'reg_lambda':
6.879400370367582e-07}. Best is trial 61 with value: 9.769342473120386.

```
[I 2023-04-16 02:54:49,168] Trial 98 finished with value:
10.807584567498479 and parameters: {'max_depth': 1, 'learning_rate':
0.017079200940730455, 'n_estimators': 409, 'min_child_weight': 2, 'gamma':
0.00013482413112321731, 'subsample': 0.50582241841654, 'colsample_bytree':
0.8878791683785946, 'reg_alpha': 1.5248475351090526e-08, 'reg_lambda':
4.931949715530416e-06}. Best is trial 61 with value: 9.769342473120386.
[I 2023-04-16 02:54:55,682] Trial 99 finished with value:
9.833721249649354 and parameters: {'max_depth': 2, 'learning_rate':
0.014949396312932359, 'n_estimators': 309, 'min_child_weight': 4, 'gamma':
0.00027540911597675387, 'subsample': 0.6082027528168207, 'colsample_bytree':
0.7312856992061668, 'reg_alpha': 6.981078161180946e-08, 'reg_lambda':
0.00013254477878179525}. Best is trial 61 with value: 9.769342473120386.
```

[ ]:
```
best = study.best_trial
newXGModel = xgb.XGBRegressor(**best.params)
newXGModel.fit(features, auto)
y_model2 = newXGModel.predict(test)
print(y_model2)
```

```
[413.8293  413.8293  413.8293  … 412.87274 413.24585 413.04507]
```

[ ]:
```
best.value
```

[ ]:
```
9.769342473120386
```

[ ]:
```
xmodel.fit(features, consul)
y_model1 = xmodel.predict(test)
print(y_model1)
```

```
[511.3286  511.3286  511.3286  … 510.11127 510.18008 509.98083]
```

[ ]:
```
xmodel.fit(features, power)
y_model3 = xmodel.predict(test)
print(y_model3)
```

```
[510.00092 510.00092 510.00092 … 509.95938 510.4253  509.69556]
```

[ ]:
```
print(y_model3.max(), y_model3.min())
```

```
508.0 504.0
```

[ ]:
```
t2new = []
for x in y_model3:
  t = round(x)
  if t == 504:
    t2new.append(413.927)
  if t == 505:
    t2new.append(416.451)
```

```
  if t == 506:
    t2new.append(418.477)
  if t == 507:
    t2new.append(420.208)
  if t == 508:
    t2new.append(421.74)
print(len(t2new))
```

```
30000
```

[ ]:
```
print(y_model2.max(), y_model2.min(), y_model2.mean())
```

```
418.1437 410.8961 412.88123
```

[ ]:
```
predict = y_model3 + y_model1 + np.array([412.88123]*30000)
ans = pd.DataFrame({"id" : testing.index, "predicted" : predict})
display(ans)
ans.to_csv("submission12.csv", index = None)
```

```
          id   predicted
0          0  1422.88123
1          1  1422.88123
2          2  1422.88123
3          3  1422.88123
4          4  1422.88123
…        …          …
29995  29995  1422.88123
29996  29996  1422.88123
29997  29997  1422.88123
29998  29998  1422.88123
29999  29999  1422.88123

[30000 rows x 2 columns]
```

[ ]:
```
print(y_model3.max(), y_model3.min())
```

```
508.0 504.0
```

[ ]:
```
!pip install pystan~=2.14
!pip install fbprophet
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting pystan~=2.14
  Downloading pystan-2.19.1.1.tar.gz (16.2 MB)
                             16.2/16.2 MB
93.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) … done
```

```
Requirement already satisfied: Cython!=0.25.1,>=0.22 in
/usr/local/lib/python3.9/dist-packages (from pystan~=2.14) (0.29.34)
Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.9/dist-
packages (from pystan~=2.14) (1.22.4)
Building wheels for collected packages: pystan
  Building wheel for pystan (setup.py) … done
  Created wheel for pystan: filename=pystan-2.19.1.1-cp39-cp39-linux_x86_64.whl
size=61826048
sha256=3f993136e7177e2bdb06efb92b6585cee24222e27a7c6cb830336d1df6eb0baa
  Stored in directory: /root/.cache/pip/wheels/b8/36/bf/7ec7e363f796373cea3eb9ea
94e83f5bbbb586d2edbf7e3417
Successfully built pystan
Installing collected packages: pystan
  Attempting uninstall: pystan
    Found existing installation: pystan 3.6.0
    Uninstalling pystan-3.6.0:
      Successfully uninstalled pystan-3.6.0
Successfully installed pystan-2.19.1.1
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Collecting fbprophet
  Using cached fbprophet-0.7.1.tar.gz (64 kB)
  Preparing metadata (setup.py) … done
Requirement already satisfied: Cython>=0.22 in /usr/local/lib/python3.9/dist-
packages (from fbprophet) (0.29.34)
Requirement already satisfied: cmdstanpy==0.9.5 in
/usr/local/lib/python3.9/dist-packages (from fbprophet) (0.9.5)
Requirement already satisfied: pystan>=2.14 in /usr/local/lib/python3.9/dist-
packages (from fbprophet) (2.19.1.1)
Requirement already satisfied: numpy>=1.15.4 in /usr/local/lib/python3.9/dist-
packages (from fbprophet) (1.22.4)
Requirement already satisfied: pandas>=1.0.4 in /usr/local/lib/python3.9/dist-
packages (from fbprophet) (1.5.3)
Requirement already satisfied: matplotlib>=2.0.0 in
/usr/local/lib/python3.9/dist-packages (from fbprophet) (3.7.1)
Requirement already satisfied: LunarCalendar>=0.0.9 in
/usr/local/lib/python3.9/dist-packages (from fbprophet) (0.0.9)
Requirement already satisfied: convertdate>=2.1.2 in
/usr/local/lib/python3.9/dist-packages (from fbprophet) (2.4.0)
Requirement already satisfied: holidays>=0.10.2 in
/usr/local/lib/python3.9/dist-packages (from fbprophet) (0.22)
Requirement already satisfied: setuptools-git>=1.2 in
/usr/local/lib/python3.9/dist-packages (from fbprophet) (1.2)
Requirement already satisfied: python-dateutil>=2.8.0 in
/usr/local/lib/python3.9/dist-packages (from fbprophet) (2.8.2)
Requirement already satisfied: tqdm>=4.36.1 in /usr/local/lib/python3.9/dist-
packages (from fbprophet) (4.65.0)
Requirement already satisfied: pymeeus<=1,>=0.3.13 in
```

```
/usr/local/lib/python3.9/dist-packages (from convertdate>=2.1.2->fbprophet)
(0.5.12)
Requirement already satisfied: korean-lunar-calendar in
/usr/local/lib/python3.9/dist-packages (from holidays>=0.10.2->fbprophet)
(0.3.1)
Requirement already satisfied: hijri-converter in /usr/local/lib/python3.9/dist-
packages (from holidays>=0.10.2->fbprophet) (2.2.4)
Requirement already satisfied: pytz in /usr/local/lib/python3.9/dist-packages
(from LunarCalendar>=0.0.9->fbprophet) (2022.7.1)
Requirement already satisfied: ephem>=3.7.5.3 in /usr/local/lib/python3.9/dist-
packages (from LunarCalendar>=0.0.9->fbprophet) (4.1.4)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=2.0.0->fbprophet)
(4.39.3)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=2.0.0->fbprophet)
(3.0.9)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.9/dist-
packages (from matplotlib>=2.0.0->fbprophet) (23.0)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=2.0.0->fbprophet)
(1.0.7)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.9/dist-
packages (from matplotlib>=2.0.0->fbprophet) (0.11.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=2.0.0->fbprophet)
(1.4.4)
Requirement already satisfied: importlib-resources>=3.2.0 in
/usr/local/lib/python3.9/dist-packages (from matplotlib>=2.0.0->fbprophet)
(5.12.0)
Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.9/dist-
packages (from matplotlib>=2.0.0->fbprophet) (8.4.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.9/dist-
packages (from python-dateutil>=2.8.0->fbprophet) (1.16.0)
Requirement already satisfied: zipp>=3.1.0 in /usr/local/lib/python3.9/dist-
packages (from importlib-resources>=3.2.0->matplotlib>=2.0.0->fbprophet)
(3.15.0)
Building wheels for collected packages: fbprophet
  Building wheel for fbprophet (setup.py) … done
  Created wheel for fbprophet: filename=fbprophet-0.7.1-py3-none-any.whl
size=9437742
sha256=1f959fc355c4d89b2432e03fbede41d09d0700dec61c40cbec4a6ecc8884507f
  Stored in directory: /root/.cache/pip/wheels/da/a4/bb/dbed5db92b2183a753dd96cc
8a56706a61484ff3959988bdaa
Successfully built fbprophet
Installing collected packages: fbprophet
Successfully installed fbprophet-0.7.1
```

[ ]:
```
from fbprophet import Prophet
```

[ ]:
```
fbtrain = pd.read_csv("train.csv")
consul = fbtrain[["dates", "TGD Consultancy Share price"]]
consul = consul.rename(columns = {"dates" : "ds", "TGD Consultancy Share price"␣
 ↪: "y"})
consul
```

[ ]:
```
                ds      y
0       1700-01-01  519.0
1       1700-01-02  518.0
2       1700-01-03  523.0
3       1700-01-04  522.0
4       1700-01-05  522.0
…              …    …
161763  2142-11-23  498.0
161764  2142-11-24  502.0
161765  2142-11-25  508.0
161766  2142-11-26  507.0
161767  2142-11-27  499.0

[161768 rows x 2 columns]
```

[ ]:
```
consul["ds"] = pd.to_datetime(consul["ds"])
```

[ ]:
```
dates = pd.Series(np.array([0]*len(consul["y"])))
print(dates)
```

```
0         0
1         0
2         0
3         0
4         0
         ..
161763    0
161764    0
161765    0
161766    0
161767    0
Length: 161768, dtype: int64
```

[ ]:
```
for i in range(len(consul["y"])):
  dates[i] = consul.iloc[:,0][i].to_pydatetime()
```

[ ]:
```
consulTrain = pd.DataFrame({"ds" : dates, "y" : consul.iloc[:,1]})
consulTrain.iloc[:,0][0]
```

[ ]:
```
datetime.datetime(1700, 1, 1, 0, 0)
```

[ ]:
```
fbmodel = Prophet(daily_seasonality = True)
fbmodel.fit(consulTrain[:10000])
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
```

[ ]:
```
<fbprophet.forecaster.Prophet at 0x7f6201fb66a0>
```

[ ]:
```
y = fbmodel.predict(consulTrain[10000:20000])
y
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
```

[ ]:

	ds	trend	yhat_lower	yhat_upper	trend_lower	\
0	1727-05-20	530.359669	494.899015	563.764868	530.359669	
1	1727-05-21	530.370044	497.274941	566.975428	530.370044	
2	1727-05-22	530.380419	494.631017	565.020871	530.380419	
3	1727-05-23	530.390794	493.554581	564.128199	530.390794	
4	1727-05-24	530.401169	495.383901	566.948982	530.401169	
…	…	…	…	…	…	
9995	1754-09-30	634.058707	-1146.711231	2530.244248	-1130.770594	
9996	1754-10-01	634.069082	-1134.440967	2508.602310	-1131.068097	
9997	1754-10-02	634.079457	-1110.850792	2504.545758	-1131.365600	
9998	1754-10-03	634.089832	-1143.342896	2531.978448	-1131.663104	
9999	1754-10-04	634.100207	-1115.016917	2516.152330	-1131.960607	

	trend_upper	additive_terms	additive_terms_lower	additive_terms_upper	\
0	530.359669	0.254353	0.254353	0.254353	
1	530.370044	0.437663	0.437663	0.437663	
2	530.380419	0.015522	0.015522	0.015522	
3	530.390794	-0.137440	-0.137440	-0.137440	
4	530.401169	-0.050338	-0.050338	-0.050338	
…	…	…	…	…	
9995	2523.317786	1.015316	1.015316	1.015316	
9996	2523.602182	0.903108	0.903108	0.903108	
9997	2523.886578	1.095479	1.095479	1.095479	

```
9998  2524.170974        0.723732              0.723732              0.723732
9999  2524.455370        0.658482              0.658482              0.658482

         daily  …    weekly  weekly_lower  weekly_upper    yearly  \
0     0.544982  …  0.100857      0.100857      0.100857 -0.391486
1     0.544982  …  0.243234      0.243234      0.243234 -0.350553
2     0.544982  … -0.176453     -0.176453     -0.176453 -0.353007
3     0.544982  … -0.287223     -0.287223     -0.287223 -0.395200
4     0.544982  … -0.122767     -0.122767     -0.122767 -0.472554
…          …  …       …             …             …       …
9995  0.544982  …  0.264504      0.264504      0.264504  0.205830
9996  0.544982  …  0.100857      0.100857      0.100857  0.257269
9997  0.544982  …  0.243234      0.243234      0.243234  0.307263
9998  0.544982  … -0.176453     -0.176453     -0.176453  0.355203
9999  0.544982  … -0.287223     -0.287223     -0.287223  0.400722

      yearly_lower  yearly_upper  multiplicative_terms  \
0        -0.391486     -0.391486                   0.0
1        -0.350553     -0.350553                   0.0
2        -0.353007     -0.353007                   0.0
3        -0.395200     -0.395200                   0.0
4        -0.472554     -0.472554                   0.0
…              …             …                   …
9995      0.205830      0.205830                   0.0
9996      0.257269      0.257269                   0.0
9997      0.307263      0.307263                   0.0
9998      0.355203      0.355203                   0.0
9999      0.400722      0.400722                   0.0

      multiplicative_terms_lower  multiplicative_terms_upper        yhat
0                            0.0                         0.0  530.614022
1                            0.0                         0.0  530.807707
2                            0.0                         0.0  530.395941
3                            0.0                         0.0  530.253354
4                            0.0                         0.0  530.350831
…                            …                         …        …
9995                         0.0                         0.0  635.074023
9996                         0.0                         0.0  634.972190
9997                         0.0                         0.0  635.174936
9998                         0.0                         0.0  634.813565
9999                         0.0                         0.0  634.758689

[10000 rows x 22 columns]
```

[ ]:
```
fbmodel = Prophet(daily_seasonality = True)
fbmodel.fit(consulTrain[:-90000])
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
```

[ ]:
```
<fbprophet.forecaster.Prophet at 0x7f625b6d1c10>
```

[ ]:
```
y1 = fbmodel.predict(consulTrain[-90000:-80000])
```

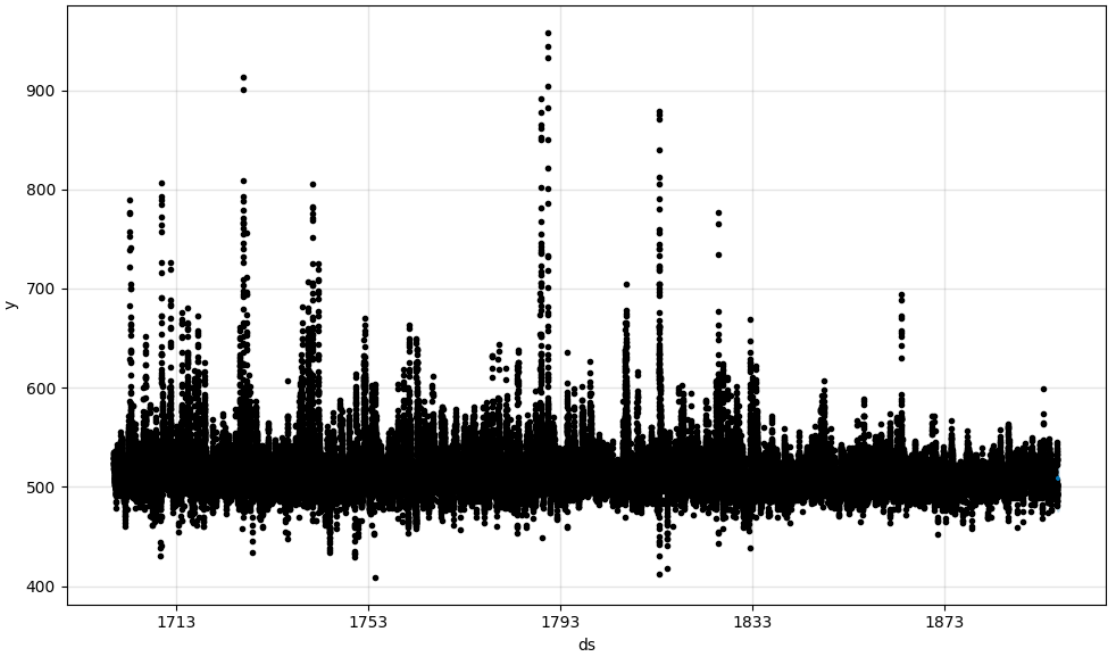
```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
```

[ ]:
```
y1
```

[ ]:
```
             ds       trend  yhat_lower  yhat_upper  trend_lower  trend_upper  \
0    1896-06-30  509.694563  477.530625  537.643929   509.694563   509.694563
1    1896-07-01  509.694488  478.280850  536.702796   509.694488   509.694488
2    1896-07-02  509.694413  479.678629  538.338749   509.694413   509.694413
3    1896-07-03  509.694338  481.565640  540.847221   509.694338   509.694338
4    1896-07-04  509.694263  478.564096  537.095001   509.694263   509.694263
…           …           …           …           …            …            …
9995 1923-11-12  508.944203  461.081874  556.563406   473.104046   547.172836
9996 1923-11-13  508.944128  461.671642  557.509435   473.100623   547.178711
9997 1923-11-14  508.944053  461.711157  556.686983   473.097200   547.184586
9998 1923-11-15  508.943978  460.225666  557.193165   473.093778   547.190461
9999 1923-11-16  508.943903  460.001888  554.668712   473.090355   547.196335

      additive_terms  additive_terms_lower  additive_terms_upper     daily  \
0          -0.910199             -0.910199             -0.910199 -1.202537
1          -0.833869             -0.833869             -0.833869 -1.202537
2          -0.845789             -0.845789             -0.845789 -1.202537
3          -0.848265             -0.848265             -0.848265 -1.202537
4          -0.802243             -0.802243             -0.802243 -1.202537
…                …                     …                     …       …
9995       -0.889060             -0.889060             -0.889060 -1.202537
9996       -1.006450             -1.006450             -1.006450 -1.202537
9997       -1.017117             -1.017117             -1.017117 -1.202537
9998       -1.103655             -1.103655             -1.103655 -1.202537
9999       -1.166213             -1.166213             -1.166213 -1.202537

      …    weekly  weekly_lower  weekly_upper    yearly  yearly_lower  \
0     …  0.014408      0.014408      0.014408  0.277930      0.277930
```

```
1     …  0.047300      0.047300      0.047300  0.321368      0.321368
2     … -0.007965     -0.007965     -0.007965  0.364712      0.364712
3     … -0.052292     -0.052292     -0.052292  0.406563      0.406563
4     … -0.045336     -0.045336     -0.045336  0.445629      0.445629
…   …       …           …           …       …           …
9995  …  0.077000      0.077000      0.077000  0.236477      0.236477
9996  …  0.014408      0.014408      0.014408  0.181680      0.181680
9997  …  0.047300      0.047300      0.047300  0.138119      0.138119
9998  … -0.007965     -0.007965     -0.007965  0.106847      0.106847
9999  … -0.052292     -0.052292     -0.052292  0.088616      0.088616

      yearly_upper  multiplicative_terms  multiplicative_terms_lower  \
0         0.277930                   0.0                         0.0
1         0.321368                   0.0                         0.0
2         0.364712                   0.0                         0.0
3         0.406563                   0.0                         0.0
4         0.445629                   0.0                         0.0
…              …                     …                           …
9995      0.236477                   0.0                         0.0
9996      0.181680                   0.0                         0.0
9997      0.138119                   0.0                         0.0
9998      0.106847                   0.0                         0.0
9999      0.088616                   0.0                         0.0

      multiplicative_terms_upper        yhat
0                            0.0  508.784364
1                            0.0  508.860619
2                            0.0  508.848624
3                            0.0  508.846072
4                            0.0  508.892019
…                            …         …
9995                         0.0  508.055143
9996                         0.0  507.937679
9997                         0.0  507.926936
9998                         0.0  507.840323
9999                         0.0  507.777690

[10000 rows x 22 columns]
```

[ ]:
```
fig = fbmodel.plot(y1[:100])
```



[ ]:
```
t = y1.iloc[:,-1]
s = consul.iloc[:, 1][-90000:-80000]
s.index = list(range(0,10000))
print(s)
print(t)
print((t - s).var()**0.5)
```

```
0       531.0
1       528.0
2       516.0
3       514.0
4       517.0
        …
9995    504.0
9996    498.0
9997    494.0
9998    490.0
9999    494.0
Name: y, Length: 10000, dtype: float64
0       508.784364
1       508.860619
2       508.848624
3       508.846072
4       508.892019
           …
9995    508.055143
```

```
9996    507.937679
9997    507.926936
9998    507.840323
9999    507.777690
Name: yhat, Length: 10000, dtype: float64
8.85332172562104
```

[ ]:
```
fbtrain = pd.read_csv("train.csv")
power = fbtrain[["dates", "TGD Power Share price"]]
power = power.rename(columns = {"dates" : "ds", "TGD Power Share price" : "y"})
power
```

[ ]:
```
                ds      y
0       1700-01-01  507.0
1       1700-01-02  507.0
2       1700-01-03  522.0
3       1700-01-04  522.0
4       1700-01-05  522.0
…              …    …
161763  2142-11-23  507.0
161764  2142-11-24  507.0
161765  2142-11-25  507.0
161766  2142-11-26  507.0
161767  2142-11-27  507.0

[161768 rows x 2 columns]
```

[ ]:
```
power["ds"] = pd.to_datetime(power["ds"])
dates = pd.Series(np.array([0]*len(power["y"])))
for i in range(len(power["y"])):
  dates[i] = power.iloc[:,0][i].to_pydatetime()
powerTrain = pd.DataFrame({"ds" : dates, "y" : power.iloc[:,1]})

fbmodel = Prophet(daily_seasonality = True)
fbmodel.fit(powerTrain[:10000])
yn = fbmodel.predict(powerTrain[10000:20000])
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
```

```
  components = components.append(new_comp)
```

[ ]:
```
t = yn.iloc[:,-1]
s = power.iloc[:, 1][10000:20000]
s.index = list(range(0,10000))
print(s)
print(t)
print((t - s).var()**0.5)
```

```
0       515.0
1       507.0
2       507.0
3       507.0
4       507.0
        …
9995    509.0
9996    509.0
9997    507.0
9998    507.0
9999    507.0
Name: y, Length: 10000, dtype: float64
0       525.723992
1       525.155600
2       525.185394
3       525.071652
4       524.961842
           …
9995    638.320768
9996    638.633664
9997    638.406352
9998    638.778707
9999    638.999208
Name: yhat, Length: 10000, dtype: float64
38.39606303230277
```

[ ]:
```
fbtrain = pd.read_csv("train.csv")
auto = fbtrain[["dates", "TGD Automobiles Share price"]]
auto = auto.rename(columns = {"dates" : "ds", "TGD Automobiles Share price" :␣
  ↪"y"})
auto



auto["ds"] = pd.to_datetime(auto["ds"])
dates = pd.Series(np.array([0]*len(auto["y"])))
for i in range(len(auto["y"])):
  dates[i] = auto.iloc[:,0][i].to_pydatetime()
autoTrain = pd.DataFrame({"ds" : dates, "y" : auto.iloc[:,1]})
```

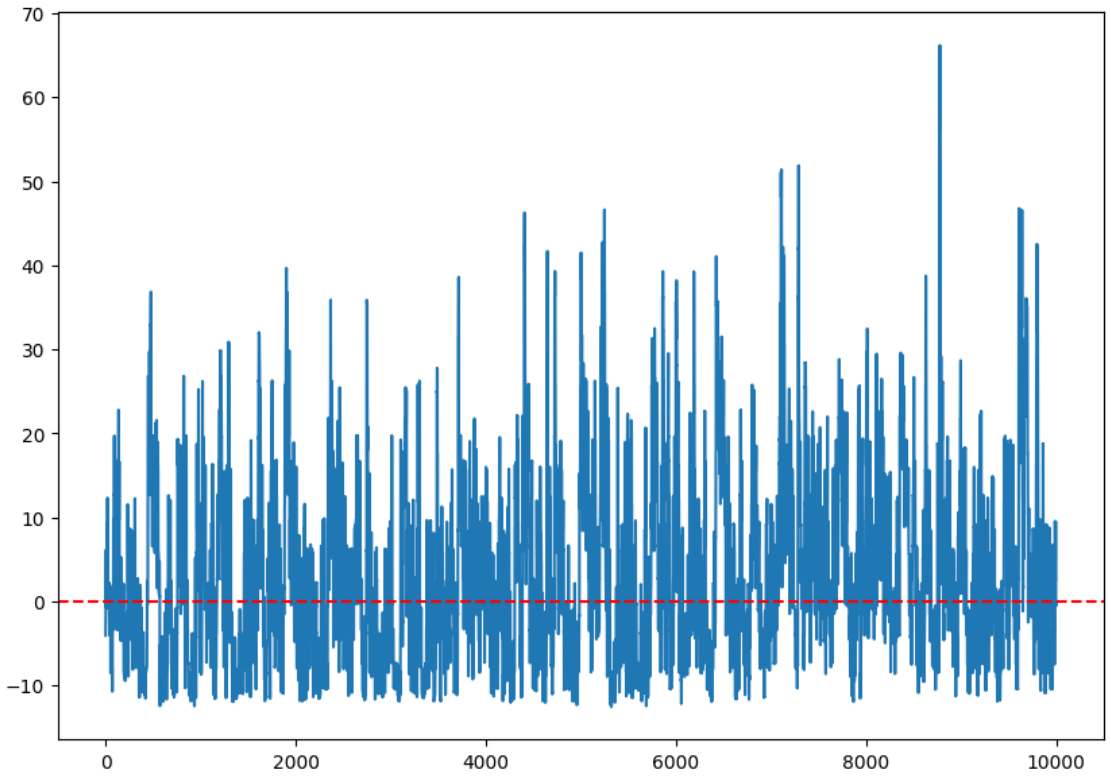
```
fbmodel = Prophet(daily_seasonality = True)
fbmodel.fit(autoTrain[:-90000])
yn = fbmodel.predict(autoTrain[-90000:-80000])

t = yn.iloc[:,-1]
s = auto.iloc[:, 1][-90000:-80000]
s.index = list(range(0,10000))
print(s)
print(t)
print((t - s).var()**0.5)
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
```

```
0       437.0
1       437.0
2       437.0
3       437.0
4       437.0
        …
9995    407.0
9996    420.0
9997    420.0
9998    420.0
9999    410.0
Name: y, Length: 10000, dtype: float64
0       416.351000
1       416.329877
2       416.323421
3       416.342775
4       416.241377
           …
9995    418.012429
9996    418.023112
9997    418.011114
9998    418.032867
9999    418.098591
```

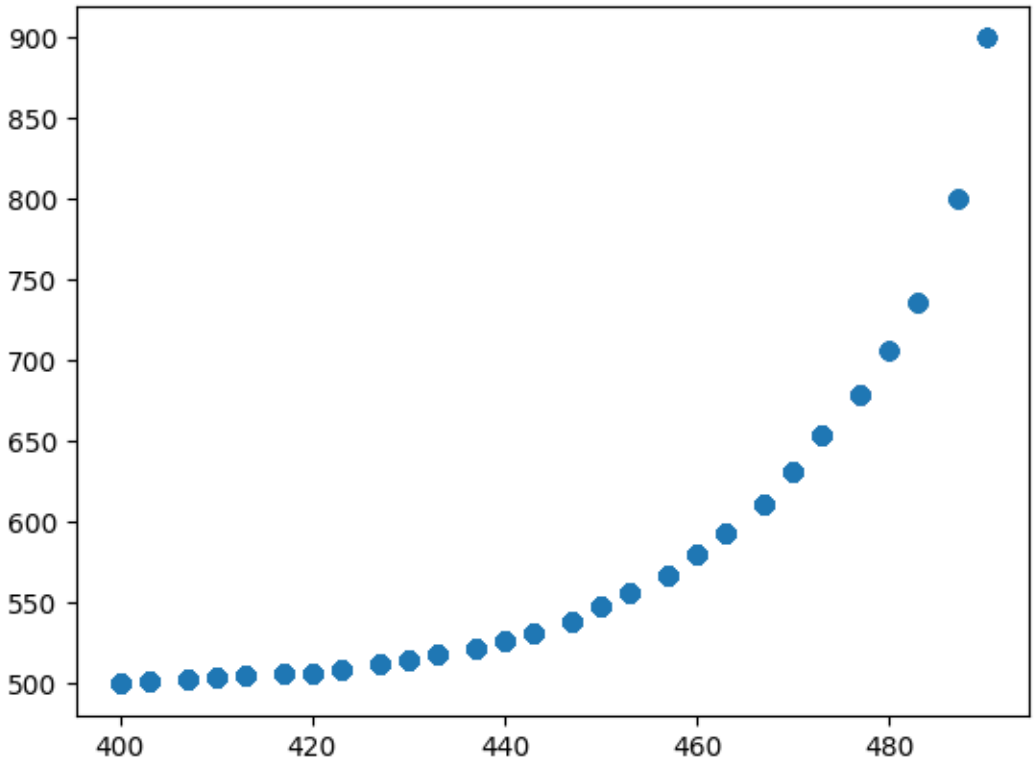
```
Name: yhat, Length: 10000, dtype: float64
10.015499803767646
```

[ ]:
```
plt.figure(figsize = (10, 7))
plt.plot(s - t)
plt.axhline(y = 0, linestyle = "--", color = "red")
plt.show()
```



[ ]:
```
plt.scatter(fbtrain.iloc[:, -2], fbtrain.iloc[:, -1])
plt.plot()
```

[ ]:
```
[]
```



[ ]:
```
newer = pd.read_csv("train.csv")
linModel = LinearRegression(fit_intercept = True)
linModel.fit(newer.iloc[:, -2][:, np.newaxis], np.log(newer.iloc[:, -1] - 435.
↪95))
```

```
<ipython-input-141-4db582a680c3>:3: FutureWarning: Support for multi-dimensional
indexing (e.g. `obj[:, None]`) is deprecated and will be removed in a future
version.  Convert to a numpy array before indexing instead.
  linModel.fit(newer.iloc[:, -2][:, np.newaxis], np.log(newer.iloc[:, -1] -
435.95))
```

[ ]:
```
LinearRegression()
```

[ ]:
```
linModel.coef_
```

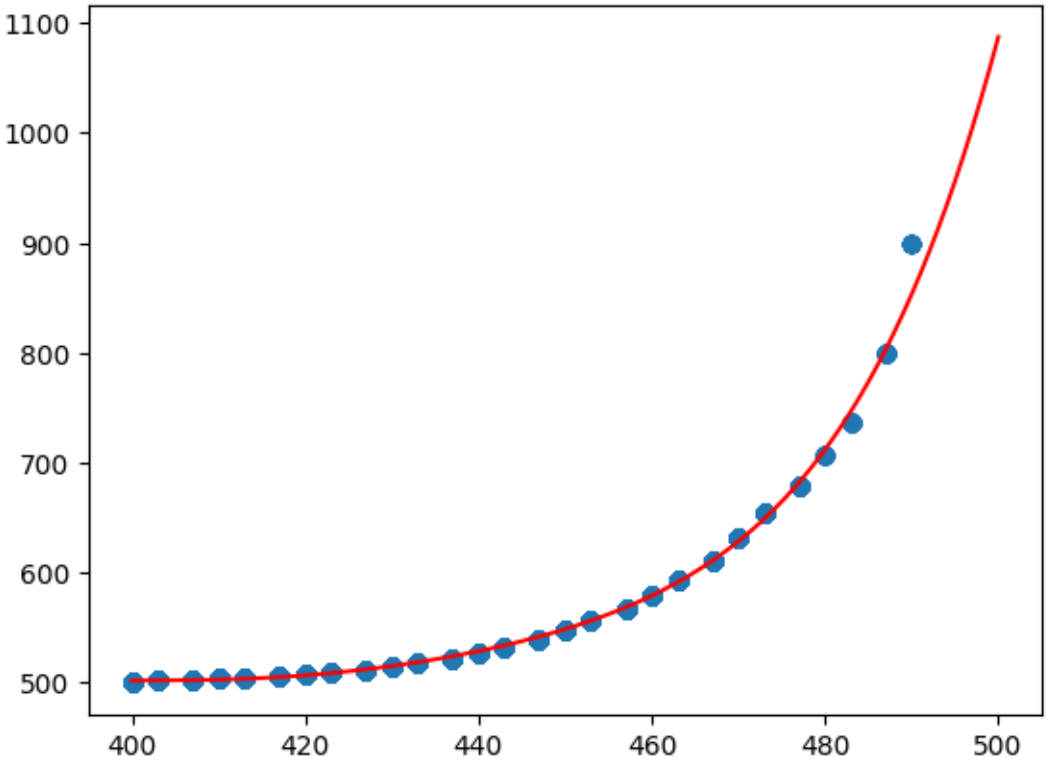
[ ]:
```
array([0.01025059])
```

[ ]:
```
linModel.intercept_
```

[ ]:
```
0.015511402360821158
```

[ ]:
```
x = np.linspace(400, 500, 1000)
y = (1.5095465806315260*10**-10)*x**6 + (-2.4287156391805662*(10**-7))*x**5 +␣
↪(1.0676216248580014*(10**-4))*x**4 + (2.5488123587843678*(10**-2))*x**3 +␣
↪(-3.5609202757077149*(10**1))*x**2 + (1.0734896229625238*(10**4))*x + (-1.
↪0916407366853179*(10**6))
```

[ ]:
```
plt.scatter(fbtrain.iloc[:, -2], fbtrain.iloc[:, -1])
plt.plot(x, y, color = "red")
plt.show()
```



[ ]:
```
fbtrain = pd.read_csv("train.csv")
auto = fbtrain[["dates", "TGD Automobiles Share price"]]
auto = auto.rename(columns = {"dates" : "ds", "TGD Automobiles Share price" :␣
↪"y"})


auto["ds"] = pd.to_datetime(auto["ds"])
dates = pd.Series(np.array([0]*len(auto["y"])))
for i in range(len(auto["y"])):
  dates[i] = auto.iloc[:,0][i].to_pydatetime()
autoTrain = pd.DataFrame({"ds" : dates, "y" : auto.iloc[:,1]})
```

[ ]:
```
fbmodel = Prophet(daily_seasonality = True)
fbmodel.fit(autoTrain[:10000])
yn = fbmodel.predict(autoTrain[10000:20000])

t = yn.iloc[:,-1]
s = auto.iloc[:, 1][10000:20000]
s.index = list(range(0,10000))
# print(s)
# print(t)
print((t - s).var()**0.5)
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
```

```
25.59187978996299
```

[ ]:
```
fbmodel = Prophet(daily_seasonality = True)
fbmodel.fit(autoTrain[10000:20000])
yn = fbmodel.predict(autoTrain[20000:30000])

t = yn.iloc[:,-1]
s = auto.iloc[:, 1][20000:30000]
s.index = list(range(0,10000))
# print(s)
# print(t)
print((t - s).var()**0.5)
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
```

```
  components = components.append(new_comp)
```

```
12.906746324825033
```

[ ]:
```
fbmodel = Prophet(daily_seasonality = True)
fbmodel.fit(autoTrain[-90000:-10000])
yn = fbmodel.predict(autoTrain[-10000:])

t = yn.iloc[:,-1]
s = auto.iloc[:, 1][-10000:]
s.index = list(range(0,10000))
# print(s)
# print(t)
print((t - s).var()**0.5)
```

```
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
/usr/local/lib/python3.9/dist-packages/fbprophet/forecaster.py:891:
FutureWarning: The frame.append method is deprecated and will be removed from
pandas in a future version. Use pandas.concat instead.
  components = components.append(new_comp)
```

```
12.542525874551801
```

[ ]:
```
testfinal = pd.read_csv("test.csv")
```