This notebook has mutiple iterations but presenting the latest one

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

In [ ]:

```
import os
os.chdir("/content/drive/My Drive/Classroom/projects/Mercari")
!ls -l
```

```
total 7772369
-rw------- 1 root root        151 Nov 19 17:35  akarshan.1711@gmail.com_CS1.gdoc
-rw------- 1 root root     192263 Jan  2 21:08 'Copy of HptTfidf2.ipynb'
-rw------- 1 root root        151 Dec 16 13:22  EDA+FE.gdoc
-rw------- 1 root root    2441752 Dec 20 16:29  EDA.ipynb
-rw------- 1 root root      14393 Dec 27 21:06  FE+prep+modelling.ipynb
-rw------- 1 root root      30163 Dec 29 18:34  HptBrnandImpute.v1.0.ipynb
-rw------- 1 root root     249493 Jan  2 20:56  HptTfidf2.ipynb
-rw------- 1 root root     117022 Jan  2 21:18  HptTfidf.ipynb
-rw------- 1 root root  117131678 Jan  1 12:07  lgbt2.csv
-rw------- 1 root root   68399264 Jan  1 02:00  lgbt3.csv
-rw------- 1 root root     927353 Dec 28 15:17  mercari_mainV2.ipynb
-rw------- 1 root root     360448 Jan  2 21:18  Mercari_to3.db
-rw------- 1 root root      77824 Jan  2 14:07  Mercari_to4.db
-rw------- 1 root root     249856 Jan  2 20:56  Mercari_to5.db
-rw------- 1 root root     196608 Jan  2 21:08  Mercari_to6.db
-rw------- 1 root root   11853944 Dec 30 21:08  price_log2.pickle
-rw------- 1 root root   11853944 Dec 31 07:52  price_log.pickle
-rw------- 1 root root      27956 Jan  2 19:51  Stack.ipynb
-rw------- 1 root root  308669128 Dec 10  2019  test_stg2.tsv.zip
-rw------- 1 root root 3474387330 Dec 30 21:08  tfidf2.pickle
-rw------- 1 root root 3623909034 Dec 30 20:50  tfidf.pickle
-rw------- 1 root root  337809843 Nov 11  2017  train.tsv
-rw------- 1 root root        272 Jan  2 19:42  Untitled
```

In [3]:

```
#importing modules/libraries
import pandas as pd
import numpy as np
import scipy
import seaborn as sns
import matplotlib.pyplot as plt
import gc
import sys
import os
import psutil
# from scipy.stats import randint as sp_randint
# from scipy.stats import uniform as sp_uniform


from tqdm.notebook import tqdm
# from collections import Counter
# from collections import defaultdict
import re
import random
# from random import sample
# from bs4 import BeautifulSoup
import pickle
import inspect
import time

import sklearn
```

```
from sklearn.feature_extraction.text import TfidfVectorizer,CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, LabelBinarizer
from sklearn.model_selection import RandomizedSearchCV
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import mean_squared_error
import lightgbm as lgb
from sklearn.linear_model import Lasso,Ridge


# import string
# # import emoji
# # from wordcloud import WordCloud
# import nltk
# nltk.download("stopwords")
# # nltk.download("brown")
# # nltk.download("names")
# # nltk.download('punkt')
# nltk.download('wordnet')
# # nltk.download('averaged_perceptron_tagger')
# # nltk.download('universal_tagset')
# # from nltk.tokenize import word_tokenize
# from nltk.corpus import stopwords
# from nltk.stem.wordnet import WordNetLemmatizer
# # from nltk.stem.porter import PorterStemmer




import warnings
warnings.filterwarnings("ignore")
```

In [4]:

```
# defining root mean square error over Log transformed y_test data
# (as linear models homoscedasticity can be kept in check for better prediction)
# and hence an effective Root Mean Square Log Error
def error(y_test, predictions):
  return np.sqrt(mean_squared_error( y_test, predictions ))
```

In [5]:

```
tr_len = 1185329# demarkation of cv data(0.8 percent)
whole_tr = 1481661# whole train data
```

In [6]:

```
# loading prepared data
with open('tfidf.pickle','rb') as f:
  df=pickle.load(f)
with open('price_log.pickle','rb') as f:
  y=pickle.load(f)
```

In [7]:

```
df = df[:whole_tr] # only taking train and cv data
gc.collect()
```

Out[7]:

```
150
```

In [8]:

```
np.isnan(df.data).sum()
```

Out[8]:

```
0
```

In [9]:

```
df.shape
```

Out[9]:

```
(1481661, 151063)
```

In [10]:

```
X_train, X_test, Y_train, Y_test = train_test_split(df,y, train_size = round(0.8*df.shape[0]))
```

In [ ]:

```
#aplha over large range
para = {'alpha': [0.001,0.01,0.1,1,2,5,10]}
clf= Ridge(max_iter=10000, tol=0.0005, solver='auto', random_state=34)
results = GridSearchCV(clf,para, cv=3, verbose=5,n_jobs=-1,scoring='neg_mean_squared_error',return_train_score=True)
results.fit(X_train,Y_train)
```

```
Fitting 3 folds for each of 7 candidates, totalling 21 fits
[Parallel(n_jobs=-1)]: Using backend LokyBackend with 2 concurrent workers.
```

In [ ]:

```
print('Best score reached: {} with params: {} '.format(results.best_score_, results.best_params_))#gscv
```

In [ ]:

```
# zooming in the range from above parameter output this cell ahs been run multiple of times
# with different values one zooming more after another
para = {'alpha': np.linspace(4,6,5)}
clf= Ridge(max_iter=10000, tol=0.0005, solver='auto', random_state=34)
results = GridSearchCV(clf,para, cv=3, verbose=5,n_jobs=-1,scoring='neg_mean_squared_error',return_train_score=True)
results.fit(X_train,Y_train)
```

In [ ]:

```
print('Best score reached: {} with params: {} '.format(results.best_score_, results.best_params_))#gscv
```

In [ ]:

```
model= results.best_estimator_
# model= Ridge(alpha=4.5, max_iter=10000, tol=0.0005, solver='auto', random_state=34)

model.fit(X_train, Y_train)
Y_pred = model.predict(X_train)

print('train error {}'.format(error(Y_train,Y_pred)))
Y_pred = model.predict(X_test)

print('test error {}'.format(error(Y_test,Y_pred)))
```

In [ ]:

```
model= results.best_estimator_

model.fit(X_train, Y_train)
Y_pred = model.predict(X_train)

print('train error {}'.format(error(Y_train,Y_pred)))
Y_pred = model.predict(X_test)

print('test error {}'.format(error(Y_test,Y_pred)))
```

```
train error 0.41566154702658537
```

test error 0.4423676809467185

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

Below cells deals with hyperparameter optimization of LGBM models. They also have been run in parallel with numerous iterations on other notebooks, but presenting the latest one here. I had earlier tried Grid search in 2 stages where first i tunned hyperparametes dealing with complexity of model like num of boosting rounds, child weight etc then tunned the convergence related parameters like learning rate and l1 and l2. That didnt work out well so i switched to optuna, Which is an alternative to sklearn's hyperopt as this has better convinience of API, visualizations, documentation and very important persistence and restarting after an instance crash of colab. Earlier i was tryin to optimize 12 to 13 hyperparametes of lgbm, but it showed results worse than ridge regression. So i went through the documenations and saw if lgbm overfits then only to hypertune all those parameters, my results were underfit. So i hypertuned only 4 to 5 parameters, and this is still going on its just one of the many copies.

In [ ]:

```
from google.colab import drive
drive.mount('/content/drive')

import os
os.chdir("/content/drive/My Drive/Classroom/projects/Mercari")
!ls -l

!pip install optuna
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount
("/content/drive", force_remount=True).
total 7772062
-rw------- 1 root root        151 Nov 19 17:35 akarshan.1711@gmail.com_CS1.gdoc
-rw------- 1 root root        151 Dec 16 13:22 EDA+FE.gdoc
-rw------- 1 root root    2441752 Dec 20 16:29 EDA.ipynb
-rw------- 1 root root      14393 Dec 27 21:06 FE+prep+modelling.ipynb
-rw------- 1 root root      30163 Dec 29 18:34 HptBrnandImpute.v1.0.ipynb
-rw------- 1 root root     226201 Dec 31 14:25 HptTfidf2.ipynb
-rw------- 1 root root      41896 Jan  2 11:30 HptTfidf.ipynb
-rw------- 1 root root  117131678 Jan  1 12:07 lgbt2.csv
-rw------- 1 root root   68399264 Jan  1 02:00 lgbt3.csv
-rw------- 1 root root     282624 Dec 31 09:07 Mercari2.db
-rw------- 1 root root     282624 Dec 31 11:18 Mercari.db
-rw------- 1 root root     182982 Dec 31 21:24 mercari_lgb_tuned.py
-rw------- 1 root root     927353 Dec 28 15:17 mercari_mainV2.ipynb
-rw------- 1 root root     118784 Jan  2 11:40 Mercari_to3.db
-rw------- 1 root root   11853944 Dec 30 21:08 price_log2.pickle
-rw------- 1 root root   11853944 Dec 31 07:52 price_log.pickle
-rw------- 1 root root      22635 Jan  2 10:18 Stack.ipynb
```

```
-rw------- 1 root root  308669128 Dec 10  2019 test_stg2.tsv.zip
-rw------- 1 root root 3474387330 Dec 30 21:08 tfidf2.pickle
-rw------- 1 root root 3623909034 Dec 30 20:50 tfidf.pickle
-rw------- 1 root root  337809843 Nov 11  2017 train.tsv
Requirement already satisfied: optuna in /usr/local/lib/python3.6/dist-packages (2.3.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.6/dist-packages (from optun
a) (4.41.1)
Requirement already satisfied: cmaes>=0.6.0 in /usr/local/lib/python3.6/dist-packages (fr
om optuna) (0.7.0)
Requirement already satisfied: numpy in /usr/local/lib/python3.6/dist-packages (from optu
na) (1.19.4)
Requirement already satisfied: alembic in /usr/local/lib/python3.6/dist-packages (from op
tuna) (1.4.3)
Requirement already satisfied: scipy!=1.4.0 in /usr/local/lib/python3.6/dist-packages (fr
om optuna) (1.4.1)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.6/dist-packages
(from optuna) (20.8)
Requirement already satisfied: cliff in /usr/local/lib/python3.6/dist-packages (from optu
na) (3.5.0)
Requirement already satisfied: colorlog in /usr/local/lib/python3.6/dist-packages (from o
ptuna) (4.6.2)
Requirement already satisfied: sqlalchemy>=1.1.0 in /usr/local/lib/python3.6/dist-package
s (from optuna) (1.3.20)
Requirement already satisfied: joblib in /usr/local/lib/python3.6/dist-packages (from opt
una) (1.0.0)
Requirement already satisfied: python-editor>=0.3 in /usr/local/lib/python3.6/dist-packag
es (from alembic->optuna) (1.0.4)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.6/dist-packages
(from alembic->optuna) (2.8.1)
Requirement already satisfied: Mako in /usr/local/lib/python3.6/dist-packages (from alemb
ic->optuna) (1.1.3)
Requirement already satisfied: pyparsing>=2.0.2 in /usr/local/lib/python3.6/dist-packages
(from packaging>=20.0->optuna) (2.4.7)
Requirement already satisfied: PyYAML>=3.12 in /usr/local/lib/python3.6/dist-packages (fr
om cliff->optuna) (3.13)
Requirement already satisfied: pbr!=2.1.0,>=2.0.0 in /usr/local/lib/python3.6/dist-packag
es (from cliff->optuna) (5.5.1)
Requirement already satisfied: PrettyTable<0.8,>=0.7.2 in /usr/local/lib/python3.6/dist-p
ackages (from cliff->optuna) (0.7.2)
Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.6/dist-packages (fro
m cliff->optuna) (1.15.0)
Requirement already satisfied: stevedore>=2.0.1 in /usr/local/lib/python3.6/dist-packages
(from cliff->optuna) (3.3.0)
Requirement already satisfied: cmd2!=0.8.3,>=0.8.0 in /usr/local/lib/python3.6/dist-packa
ges (from cliff->optuna) (1.4.0)
Requirement already satisfied: MarkupSafe>=0.9.2 in /usr/local/lib/python3.6/dist-package
s (from Mako->alembic->optuna) (1.1.1)
Requirement already satisfied: importlib-metadata>=1.7.0; python_version < "3.8" in /usr/
local/lib/python3.6/dist-packages (from stevedore>=2.0.1->cliff->optuna) (3.3.0)
Requirement already satisfied: colorama>=0.3.7 in /usr/local/lib/python3.6/dist-packages
(from cmd2!=0.8.3,>=0.8.0->cliff->optuna) (0.4.4)
Requirement already satisfied: attrs>=16.3.0 in /usr/local/lib/python3.6/dist-packages (f
rom cmd2!=0.8.3,>=0.8.0->cliff->optuna) (20.3.0)
Requirement already satisfied: pyperclip>=1.6 in /usr/local/lib/python3.6/dist-packages (
from cmd2!=0.8.3,>=0.8.0->cliff->optuna) (1.8.1)
Requirement already satisfied: wcwidth>=0.1.7 in /usr/local/lib/python3.6/dist-packages (
from cmd2!=0.8.3,>=0.8.0->cliff->optuna) (0.2.5)
Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.6/dist-packages (from
importlib-metadata>=1.7.0; python_version < "3.8"->stevedore>=2.0.1->cliff->optuna) (3.4.
0)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in /usr/l
ocal/lib/python3.6/dist-packages (from importlib-metadata>=1.7.0; python_version < "3.8"-
>stevedore>=2.0.1->cliff->optuna) (3.7.4.3)
```

In [ ]:

```
#https://optuna.readthedocs.io/en/stable/index.html

import lightgbm as lgb
import numpy as np
import sklearn.datasets
```

```
import sklearn.metrics
from sklearn.model_selection import train_test_split
import pickle

import optuna



def obj_func(trial,data, target):

    param = {
        "objective": "regression",
        "metric": "rmse",
        "verbosity": -1,
        "boosting_type": "gbdt",
        "n_estimator":trial.suggest_int("n_estimator", 80, 2500),
        "max_depth": trial.suggest_int("max_depth", 10,40),
        "num_leaves": trial.suggest_int("num_leaves", 80, 200),
        # "min_child_samples": trial.suggest_int("min_child_samples", 50, 300),
        # "min_child_weight" : trial.suggest_float('min_child_weight', 0.002, 1.0),
        "subsample_for_bin":trial.suggest_int("subsample_for_bin", 50000, 100000),
        "learning_rate" : trial.suggest_loguniform('learning_rate', 1e-1, 5),
        # "lambda_l1": trial.suggest_loguniform("lambda_l1", 1e-5, 10.0),
        # "lambda_l2": trial.suggest_loguniform("lambda_l2", 1e-5, 10.0),
        # "feature_fraction": trial.suggest_float("feature_fraction", 0.2, 0.85),
        # "bagging_fraction": trial.suggest_float("bagging_fraction", 0.2, 0.85),
        # "bagging_freq": trial.suggest_int("bagging_freq", 1, 7),


    }
    tr_len= 1185329
    train_x, valid_x, train_y, valid_y = train_test_split(
                                        data,target,train_size = tr_len
                                        )
    dtrain = lgb.Dataset(train_x, label=train_y)
    dvalid = lgb.Dataset(valid_x, label=valid_y)




    # Add a callback for pruning.
    pruning_callback = optuna.integration.LightGBMPruningCallback(trial, "rmse")
    gbm = lgb.train(
        param, dtrain, valid_sets=[dvalid], verbose_eval=False, callbacks=[pruning_callb
ack]
    )

    preds = gbm.predict(valid_x)
    pred_labels = np.rint(preds)


    return np.sqrt(sklearn.metrics.mean_squared_error(valid_y, pred_labels))


if __name__ == "__main__":

  with open('tfidf.pickle','rb') as f:
    data=pickle.load(f)
  with open('price_log.pickle','rb') as f:
    target=pickle.load(f)


    whole_len = 1481661
    data = data[:whole_len,:]
    target = target[:whole_len]


    study_name = 'Mercari'
    study = optuna.create_study(study_name=study_name,
                                storage='sqlite:///Mercari_to3.db',
                                load_if_exists=True,
                                pruner=optuna.pruners.MedianPruner(n_warmup_steps=10), d
```

```
irection="minimize"
                                )


    study.optimize(lambda trial : obj_func(trial,data, target), n_trials=20 ,gc_after_tr
ial=True)

    print("Number of finished trials: {}".format(len(study.trials)))

    print("Best trial:")
    trial = study.best_trial

    print("  Value: {}".format(trial.value))

    print("  Params: ")
    for key, value in trial.params.items():
        print("    {}: {}".format(key, value))
```

```
[I 2021-01-02 11:42:11,987] Using an existing study with name 'Mercari' instead of creati
ng a new one.
[I 2021-01-02 11:52:06,905] Trial 13 finished with value: 0.550835330019086 and parameter
s: {'n_estimator': 290, 'max_depth': 30, 'num_leaves': 192, 'subsample_for_bin': 86578, '
learning_rate': 0.7556419396442797}. Best is trial 0 with value: 0.5502095808113284.
[I 2021-01-02 11:58:35,533] Trial 14 pruned. Trial was pruned at iteration 63.
[I 2021-01-02 12:00:33,854] Trial 15 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 12:07:39,048] Trial 16 finished with value: 0.5470090793492802 and paramete
rs: {'n_estimator': 337, 'max_depth': 18, 'num_leaves': 184, 'subsample_for_bin': 83326,
'learning_rate': 0.4398638878482992}. Best is trial 16 with value: 0.5470090793492802.
[I 2021-01-02 12:13:17,393] Trial 17 finished with value: 0.5520590552491569 and paramete
rs: {'n_estimator': 1076, 'max_depth': 15, 'num_leaves': 98, 'subsample_for_bin': 79062,
'learning_rate': 0.37944470650238377}. Best is trial 16 with value: 0.5470090793492802.
[I 2021-01-02 12:14:56,642] Trial 18 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 12:16:49,806] Trial 19 pruned. Trial was pruned at iteration 16.
[I 2021-01-02 12:18:44,244] Trial 20 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 12:24:44,523] Trial 21 finished with value: 0.55022269825975 and parameters
: {'n_estimator': 442, 'max_depth': 14, 'num_leaves': 141, 'subsample_for_bin': 91645, 'l
earning_rate': 0.4007428419820709}. Best is trial 16 with value: 0.5470090793492802.
[I 2021-01-02 12:30:39,609] Trial 22 finished with value: 0.5504292829382879 and paramete
rs: {'n_estimator': 478, 'max_depth': 14, 'num_leaves': 144, 'subsample_for_bin': 91967,
'learning_rate': 0.4285084549757989}. Best is trial 16 with value: 0.5470090793492802.
[I 2021-01-02 12:32:13,140] Trial 23 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 12:33:39,881] Trial 24 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 12:41:15,966] Trial 25 finished with value: 0.5475869193220357 and paramete
rs: {'n_estimator': 828, 'max_depth': 22, 'num_leaves': 109, 'subsample_for_bin': 94819,
'learning_rate': 0.5512151140452101}. Best is trial 16 with value: 0.5470090793492802.
[I 2021-01-02 12:42:46,140] Trial 26 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 12:50:31,745] Trial 27 finished with value: 0.5457000599074706 and paramete
rs: {'n_estimator': 1143, 'max_depth': 25, 'num_leaves': 93, 'subsample_for_bin': 84098,
'learning_rate': 0.6442645415263681}. Best is trial 27 with value: 0.5457000599074706.
[I 2021-01-02 12:52:02,238] Trial 28 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:00:16,836] Trial 29 finished with value: 0.5439879938747317 and paramete
rs: {'n_estimator': 1498, 'max_depth': 27, 'num_leaves': 110, 'subsample_for_bin': 77253,
'learning_rate': 0.49373962887549777}. Best is trial 29 with value: 0.5439879938747317.
[I 2021-01-02 13:01:42,307] Trial 30 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:03:14,466] Trial 31 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:10:45,946] Trial 32 finished with value: 0.545716022503374 and parameter
s: {'n_estimator': 1385, 'max_depth': 23, 'num_leaves': 103, 'subsample_for_bin': 85419,
'learning_rate': 0.5578566183389}. Best is trial 29 with value: 0.5439879938747317.
[I 2021-01-02 13:18:49,946] Trial 33 finished with value: 0.5455415910135475 and paramete
rs: {'n_estimator': 1481, 'max_depth': 27, 'num_leaves': 89, 'subsample_for_bin': 86090,
'learning_rate': 0.5706309230126199}. Best is trial 29 with value: 0.5439879938747317.
[I 2021-01-02 13:27:02,890] Trial 34 finished with value: 0.5474066555038809 and paramete
rs: {'n_estimator': 1379, 'max_depth': 28, 'num_leaves': 91, 'subsample_for_bin': 86327,
'learning_rate': 0.6494243713250953}. Best is trial 29 with value: 0.5439879938747317.
[I 2021-01-02 13:28:36,717] Trial 35 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:37:48,631] Trial 36 finished with value: 0.5429145723272504 and paramete
rs: {'n_estimator': 1525, 'max_depth': 32, 'num_leaves': 88, 'subsample_for_bin': 88234,
'learning_rate': 0.5653290509665294}. Best is trial 36 with value: 0.5429145723272504.
[I 2021-01-02 13:39:10,468] Trial 37 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:40:40,696] Trial 38 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:42:06,719] Trial 39 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:43:38,948] Trial 40 pruned. Trial was pruned at iteration 10.
```

```
[I 2021-01-02 13:45:10,048] Trial 41 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 13:52:46,322] Trial 42 finished with value: 0.5454639467422769 and parameters: {'n_estimator': 1326, 'max_depth': 24, 'num_leaves': 102, 'subsample_for_bin': 85435, 'learning_rate': 0.548554548379258}. Best is trial 36 with value: 0.5429145723272504.
[I 2021-01-02 14:01:33,103] Trial 43 finished with value: 0.5437206693697554 and parameters: {'n_estimator': 1742, 'max_depth': 29, 'num_leaves': 111, 'subsample_for_bin': 81707, 'learning_rate': 0.5238254427166154}. Best is trial 36 with value: 0.5429145723272504.
[I 2021-01-02 14:03:33,092] Trial 44 pruned. Trial was pruned at iteration 14.
[I 2021-01-02 14:05:16,250] Trial 45 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 14:14:35,526] Trial 46 finished with value: 0.5431045127644969 and parameters: {'n_estimator': 1509, 'max_depth': 34, 'num_leaves': 100, 'subsample_for_bin': 80347, 'learning_rate': 0.5502505339493712}. Best is trial 36 with value: 0.5429145723272504.
[I 2021-01-02 14:16:09,687] Trial 47 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 14:18:44,100] Trial 48 pruned. Trial was pruned at iteration 18.
[I 2021-01-02 14:27:51,835] Trial 49 finished with value: 0.5416196338607705 and parameters: {'n_estimator': 1298, 'max_depth': 31, 'num_leaves': 101, 'subsample_for_bin': 82534, 'learning_rate': 0.49559122346843787}. Best is trial 49 with value: 0.5416196338607705.
[I 2021-01-02 14:33:27,734] Trial 50 pruned. Trial was pruned at iteration 68.
[I 2021-01-02 14:35:09,347] Trial 51 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 14:44:42,747] Trial 52 finished with value: 0.5423186383544838 and parameters: {'n_estimator': 1344, 'max_depth': 34, 'num_leaves': 101, 'subsample_for_bin': 78844, 'learning_rate': 0.4930522194989849}. Best is trial 49 with value: 0.5416196338607705.
[I 2021-01-02 14:46:15,486] Trial 53 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 14:48:01,497] Trial 54 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 14:49:31,374] Trial 55 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 14:59:05,865] Trial 56 finished with value: 0.5452957504834804 and parameters: {'n_estimator': 950, 'max_depth': 36, 'num_leaves': 85, 'subsample_for_bin': 82612, 'learning_rate': 0.7310280109001812}. Best is trial 49 with value: 0.5416196338607705.
[I 2021-01-02 15:00:31,787] Trial 57 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 15:02:12,857] Trial 58 pruned. Trial was pruned at iteration 11.
[I 2021-01-02 15:03:47,547] Trial 59 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 15:12:42,064] Trial 60 finished with value: 0.5430282632052358 and parameters: {'n_estimator': 2233, 'max_depth': 29, 'num_leaves': 116, 'subsample_for_bin': 88246, 'learning_rate': 0.5112979409905812}. Best is trial 49 with value: 0.5416196338607705.
[I 2021-01-02 15:14:18,490] Trial 61 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 15:20:11,346] Trial 62 pruned. Trial was pruned at iteration 61.
```

```
Number of finished trials: 63
Best trial:
  Value: 0.5416196338607705
  Params:
    learning_rate: 0.49559122346843787
    max_depth: 31
    n_estimator: 1298
    num_leaves: 101
    subsample_for_bin: 82534
```

In [ ]:

```
#https://optuna.readthedocs.io/en/stable/index.html

import lightgbm as lgb
import numpy as np
import sklearn.datasets
import sklearn.metrics
from sklearn.model_selection import train_test_split
import pickle

import optuna



def obj_func(trial,data, target):

    param = {
        "objective": "regression",
        "metric": "rmse",
        "verbosity": -1,
        "boosting_type": "gbdt",
        "n_estimator":trial.suggest_int("n_estimator", 2000, 2500),
        "max_depth": trial.suggest_int("max_depth", 30,42),
        "num_leaves": trial.suggest_int("num_leaves", 120, 200),
```

```
        # "min_child_samples": trial.suggest_int("min_child_samples", 50, 300),
        # "min_child_weight" : trial.suggest_float('min_child_weight', 0.002, 1.0),
        "subsample_for_bin":trial.suggest_int("subsample_for_bin", 50000, 100000),
        "learning_rate" : trial.suggest_loguniform('learning_rate', 1e-1, 1),
        # "lambda_l1": trial.suggest_loguniform("lambda_l1", 1e-5, 10.0),
        # "lambda_l2": trial.suggest_loguniform("lambda_l2", 1e-5, 10.0),
        # "feature_fraction": trial.suggest_float("feature_fraction", 0.2, 0.85),
        # "bagging_fraction": trial.suggest_float("bagging_fraction", 0.2, 0.85),
        # "bagging_freq": trial.suggest_int("bagging_freq", 1, 7),


    }
    tr_len= 1185329
    train_x, valid_x, train_y, valid_y = train_test_split(
                                        data,target,train_size = tr_len
                                        )
    dtrain = lgb.Dataset(train_x, label=train_y)
    dvalid = lgb.Dataset(valid_x, label=valid_y)




    # Add a callback for pruning.
    pruning_callback = optuna.integration.LightGBMPruningCallback(trial, "rmse")
    gbm = lgb.train(
        param, dtrain, valid_sets=[dvalid], verbose_eval=False, callbacks=[pruning_callb
ack]
    )

    preds = gbm.predict(valid_x)
    pred_labels = np.rint(preds)


    return np.sqrt(sklearn.metrics.mean_squared_error(valid_y, pred_labels))


if __name__ == "__main__":

  with open('tfidf.pickle','rb') as f:
    data=pickle.load(f)
  with open('price_log.pickle','rb') as f:
    target=pickle.load(f)


    whole_len = 1481661
    data = data[:whole_len,:]
    target = target[:whole_len]


    study_name = 'Mercari'
    study = optuna.create_study(study_name=study_name,
                                storage='sqlite:///Mercari_to3.db',
                                load_if_exists=True,
                                pruner=optuna.pruners.MedianPruner(n_warmup_steps=10), d
irection="minimize"
                                )


    study.optimize(lambda trial : obj_func(trial,data, target), n_trials=20 ,gc_after_tr
ial=True)

    print("Number of finished trials: {}".format(len(study.trials)))

    print("Best trial:")
    trial = study.best_trial

    print("  Value: {}".format(trial.value))

    print("  Params: ")
    for key, value in trial.params.items():
        print("    {}: {}".format(key, value))
```

```
[I 2021-01-02 15:28:09,390] Using an existing study with name 'Mercari' instead of creati
ng a new one.
[I 2021-01-02 15:37:26,550] Trial 63 finished with value: 0.5412700356918323 and paramete
rs: {'n_estimator': 2346, 'max_depth': 30, 'num_leaves': 122, 'subsample_for_bin': 84144,
'learning_rate': 0.5232593809983801}. Best is trial 63 with value: 0.5412700356918323.
[I 2021-01-02 15:46:45,782] Trial 64 finished with value: 0.5410045171521921 and paramete
rs: {'n_estimator': 2372, 'max_depth': 30, 'num_leaves': 123, 'subsample_for_bin': 83697,
'learning_rate': 0.4917508722758207}. Best is trial 64 with value: 0.5410045171521921.
[I 2021-01-02 15:55:56,560] Trial 65 finished with value: 0.5451074998650021 and paramete
rs: {'n_estimator': 2369, 'max_depth': 30, 'num_leaves': 123, 'subsample_for_bin': 84198,
'learning_rate': 0.6294799909717015}. Best is trial 64 with value: 0.5410045171521921.
[I 2021-01-02 16:05:43,884] Trial 66 finished with value: 0.5413544491340004 and paramete
rs: {'n_estimator': 2346, 'max_depth': 31, 'num_leaves': 137, 'subsample_for_bin': 88081,
'learning_rate': 0.4871327052245831}. Best is trial 64 with value: 0.5410045171521921.
[I 2021-01-02 16:15:24,586] Trial 67 finished with value: 0.5422780547002413 and paramete
rs: {'n_estimator': 2342, 'max_depth': 30, 'num_leaves': 126, 'subsample_for_bin': 93921,
'learning_rate': 0.48184038751272507}. Best is trial 64 with value: 0.5410045171521921.
[I 2021-01-02 16:25:12,995] Trial 68 finished with value: 0.5420539025219673 and paramete
rs: {'n_estimator': 2345, 'max_depth': 30, 'num_leaves': 137, 'subsample_for_bin': 95123,
'learning_rate': 0.4776114468249519}. Best is trial 64 with value: 0.5410045171521921.
[I 2021-01-02 16:35:02,454] Trial 69 finished with value: 0.5410729925394048 and paramete
rs: {'n_estimator': 2346, 'max_depth': 30, 'num_leaves': 138, 'subsample_for_bin': 98771,
'learning_rate': 0.48134652093192926}. Best is trial 64 with value: 0.5410045171521921.
[I 2021-01-02 16:36:53,868] Trial 70 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 16:47:09,881] Trial 71 finished with value: 0.5403134396525309 and paramete
rs: {'n_estimator': 2343, 'max_depth': 31, 'num_leaves': 149, 'subsample_for_bin': 93718,
'learning_rate': 0.4104154648134682}. Best is trial 71 with value: 0.5403134396525309.
[I 2021-01-02 16:57:22,306] Trial 72 finished with value: 0.5397973639219291 and paramete
rs: {'n_estimator': 2336, 'max_depth': 31, 'num_leaves': 150, 'subsample_for_bin': 94378,
'learning_rate': 0.4196182357650816}. Best is trial 72 with value: 0.5397973639219291.
[I 2021-01-02 17:08:00,099] Trial 73 finished with value: 0.5390072060426787 and paramete
rs: {'n_estimator': 2294, 'max_depth': 31, 'num_leaves': 151, 'subsample_for_bin': 96101,
'learning_rate': 0.4168916591930659}. Best is trial 73 with value: 0.5390072060426787.
[I 2021-01-02 17:18:28,874] Trial 74 finished with value: 0.5401474777801137 and paramete
rs: {'n_estimator': 2295, 'max_depth': 31, 'num_leaves': 155, 'subsample_for_bin': 98165,
'learning_rate': 0.4064035119704925}. Best is trial 73 with value: 0.5390072060426787.
[I 2021-01-02 17:28:57,744] Trial 75 finished with value: 0.5415070534387871 and paramete
rs: {'n_estimator': 2287, 'max_depth': 31, 'num_leaves': 155, 'subsample_for_bin': 98424,
'learning_rate': 0.41256823623795713}. Best is trial 73 with value: 0.5390072060426787.
[I 2021-01-02 17:30:57,108] Trial 76 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 17:41:20,446] Trial 77 finished with value: 0.541395789853596 and parameter
s: {'n_estimator': 2388, 'max_depth': 32, 'num_leaves': 149, 'subsample_for_bin': 97389,
'learning_rate': 0.40642347179042704}. Best is trial 73 with value: 0.5390072060426787.
[I 2021-01-02 17:51:19,843] Trial 78 finished with value: 0.5401253662174467 and paramete
rs: {'n_estimator': 2314, 'max_depth': 30, 'num_leaves': 160, 'subsample_for_bin': 91309,
'learning_rate': 0.45470291979199096}. Best is trial 73 with value: 0.5390072060426787.
[I 2021-01-02 17:53:16,041] Trial 79 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 18:03:27,237] Trial 80 finished with value: 0.5428828951613501 and paramete
rs: {'n_estimator': 2255, 'max_depth': 30, 'num_leaves': 164, 'subsample_for_bin': 91442,
'learning_rate': 0.6106074583995795}. Best is trial 73 with value: 0.5390072060426787.
[I 2021-01-02 18:05:21,999] Trial 81 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 18:15:28,260] Trial 82 finished with value: 0.541391517523529 and parameter
s: {'n_estimator': 2324, 'max_depth': 30, 'num_leaves': 159, 'subsample_for_bin': 91032,
'learning_rate': 0.4489568547185416}. Best is trial 73 with value: 0.5390072060426787.
```

```
Number of finished trials: 83
Best trial:
  Value: 0.5390072060426787
  Params:
    learning_rate: 0.4168916591930659
    max_depth: 31
    n_estimator: 2294
    num_leaves: 151
    subsample_for_bin: 96101
```

In [ ]:

```
optuna.visualization.plot_optimization_history(study)
```

In [ ]:

```
optuna.visualization.plot_slice(study)
```

In [ ]:

```
optuna.visualization.plot_contour(study, params=['n_estimator', 'learning_rate'])
```

This cell is still running.

In [ ]:

```
#https://optuna.readthedocs.io/en/stable/index.html

import lightgbm as lgb
import numpy as np
import sklearn.datasets
import sklearn.metrics
from sklearn.model_selection import train_test_split
import pickle

import optuna


def obj_func(trial,data, target):

    param = {
        "objective": "regression",
        "metric": "rmse",
        "verbosity": -1,
        "boosting_type": "gbdt",
        "n_estimator":trial.suggest_int("n_estimator", 2000, 2500),
        "max_depth": trial.suggest_int("max_depth", 30,42),
        "num_leaves": trial.suggest_int("num_leaves", 120, 200),
        # "min_child_samples": trial.suggest_int("min_child_samples", 50, 300),
        # "min_child_weight" : trial.suggest_float('min_child_weight', 0.002, 1.0),
        "subsample_for_bin":trial.suggest_int("subsample_for_bin", 50000, 100000),
        "learning_rate" : trial.suggest_loguniform('learning_rate', 0.1, 0.5),
        # "lambda_l1": trial.suggest_loguniform("lambda_l1", 1e-5, 10.0),
        # "lambda_l2": trial.suggest_loguniform("lambda_l2", 1e-5, 10.0),
        # "feature_fraction": trial.suggest_float("feature_fraction", 0.2, 0.85),
        # "bagging_fraction": trial.suggest_float("bagging_fraction", 0.2, 0.85),
        # "bagging_freq": trial.suggest_int("bagging_freq", 1, 7),


    }
    tr_len= 1185329
    train_x, valid_x, train_y, valid_y = train_test_split(
```

```
                                    data,target,train_size = tr_len
                                    )
    dtrain = lgb.Dataset(train_x, label=train_y)
    dvalid = lgb.Dataset(valid_x, label=valid_y)




    # Add a callback for pruning.
    pruning_callback = optuna.integration.LightGBMPruningCallback(trial, "rmse")
    gbm = lgb.train(
        param, dtrain, valid_sets=[dvalid], verbose_eval=False, callbacks=[pruning_callb
ack]
    )

    preds = gbm.predict(valid_x)
    pred_labels = np.rint(preds)


    return np.sqrt(sklearn.metrics.mean_squared_error(valid_y, pred_labels))


if __name__ == "__main__":

  with open('tfidf.pickle','rb') as f:
    data=pickle.load(f)
  with open('price_log.pickle','rb') as f:
    target=pickle.load(f)


    whole_len = 1481661
    data = data[:whole_len,:]
    target = target[:whole_len]


    study_name = 'Mercari'
    study = optuna.create_study(study_name=study_name,
                                storage='sqlite:///Mercari_to3.db',
                                load_if_exists=True,
                                pruner=optuna.pruners.MedianPruner(n_warmup_steps=10), d
irection="minimize"
                                )


    study.optimize(lambda trial : obj_func(trial,data, target), n_trials=20 ,gc_after_tr
ial=True)

    print("Number of finished trials: {}".format(len(study.trials)))

    print("Best trial:")
    trial = study.best_trial

    print("  Value: {}".format(trial.value))

    print("  Params: ")
    for key, value in trial.params.items():
        print("    {}: {}".format(key, value))
```

```
[I 2021-01-02 18:29:21,802] Using an existing study with name 'Mercari' instead of creati
ng a new one.
[I 2021-01-02 18:42:02,025] Trial 83 finished with value: 0.5377291210244696 and paramete
rs: {'n_estimator': 2368, 'max_depth': 42, 'num_leaves': 151, 'subsample_for_bin': 99755,
'learning_rate': 0.4668527455862786}. Best is trial 83 with value: 0.5377291210244696.
[I 2021-01-02 18:52:28,839] Trial 84 finished with value: 0.5397384130004586 and paramete
rs: {'n_estimator': 2369, 'max_depth': 32, 'num_leaves': 151, 'subsample_for_bin': 96346,
'learning_rate': 0.46649778950467075}. Best is trial 83 with value: 0.5377291210244696.
[I 2021-01-02 19:04:58,556] Trial 85 finished with value: 0.5369570866388655 and paramete
rs: {'n_estimator': 2366, 'max_depth': 42, 'num_leaves': 152, 'subsample_for_bin': 99916,
'learning_rate': 0.4235203749209467}. Best is trial 85 with value: 0.5369570866388655.
[I 2021-01-02 19:17:33,639] Trial 86 finished with value: 0.5381498850308358 and paramete
rs: {'n_estimator': 2435, 'max_depth': 42, 'num_leaves': 151, 'subsample_for_bin': 96439,
'learning_rate': 0.42027586074217377}. Best is trial 85 with value: 0.5369570866388655.
```

```
[I 2021-01-02 19:30:14,198] Trial 87 finished with value: 0.5360534149517756 and parameters: {'n_estimator': 2457, 'max_depth': 42, 'num_leaves': 151, 'subsample_for_bin': 99947, 'learning_rate': 0.42742671823862655}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 19:43:12,922] Trial 88 finished with value: 0.5382915609284268 and parameters: {'n_estimator': 2460, 'max_depth': 42, 'num_leaves': 152, 'subsample_for_bin': 99808, 'learning_rate': 0.3858588561284566}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 19:55:34,969] Trial 89 finished with value: 0.5373962762290251 and parameters: {'n_estimator': 2465, 'max_depth': 42, 'num_leaves': 152, 'subsample_for_bin': 99840, 'learning_rate': 0.4319161764254474}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 19:57:29,415] Trial 90 pruned. Trial was pruned at iteration 10.
[I 2021-01-02 20:10:00,876] Trial 91 finished with value: 0.5380590301987365 and parameters: {'n_estimator': 2441, 'max_depth': 42, 'num_leaves': 144, 'subsample_for_bin': 96631, 'learning_rate': 0.4258668841171013}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 20:22:28,573] Trial 92 finished with value: 0.5371775391765338 and parameters: {'n_estimator': 2452, 'max_depth': 42, 'num_leaves': 143, 'subsample_for_bin': 96797, 'learning_rate': 0.4263915458645329}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 20:34:59,228] Trial 93 finished with value: 0.5392662550432215 and parameters: {'n_estimator': 2428, 'max_depth': 42, 'num_leaves': 146, 'subsample_for_bin': 96595, 'learning_rate': 0.466231366393573}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 20:47:36,380] Trial 94 finished with value: 0.5384087420595225 and parameters: {'n_estimator': 2444, 'max_depth': 42, 'num_leaves': 143, 'subsample_for_bin': 99629, 'learning_rate': 0.4313682218539111}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 20:59:50,578] Trial 95 finished with value: 0.5364748001106293 and parameters: {'n_estimator': 2451, 'max_depth': 41, 'num_leaves': 144, 'subsample_for_bin': 99574, 'learning_rate': 0.4259125308215095}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 21:11:50,417] Trial 96 finished with value: 0.5381116502780182 and parameters: {'n_estimator': 2453, 'max_depth': 41, 'num_leaves': 142, 'subsample_for_bin': 99970, 'learning_rate': 0.4300190059887762}. Best is trial 87 with value: 0.5360534149517756.
[I 2021-01-02 21:13:44,982] Trial 97 pruned. Trial was pruned at iteration 10.
```

In [ ]:

```
optuna.visualization.plot_optimization_history(study)
```

In [ ]:

```
optuna.visualization.plot_slice(study)
```

In [ ]:

```
optuna.visualization.plot_contour(study, params=['n_estimator', 'learning_rate'])
```