The zip file contains following python files:

1) trainer.py
   This file contains training functions for catboost and lightgbm. Also contains few functions that will be used for generating output.
2) data_transformer.py
   This contains datacleaner class that are collection of datacleaning and data transforming utils. Most of them can be used as annotations and few are functions are having specific task.
3) model_runner.py
   This is the main file that will be used to design the datacleaning , datatransforming , training and output generation.
4) tox_train.ini
   This file contains parameters that will be used in datacleaning, creating models, training and output generation. [For training only]
5) tox.ini
   This file contains parameters that will be used in datacleaning, creating models, training and output generation. [For predi
6) catboost_list.pkl
   This file contains names of columns/features on which the catboost pretrained model is trained.
7) lgbm_list.pkl
   This file contains names of columns/features on which the catboost pretrained model is trained.
8) submission.csv
   This file contained the result for 20 % of the train set - dataset_00_with_header.csv
9) dataset_00_with_header.csv:
   Data on which model was trained.

Pre-requisites:

1) Python 3.6.
2) Latest catboost, lightgbm (recommended to install from conda forge)
3) Latest Scikit-learn, pandas, numpy
4) I think all other modules are installed by default. If any other module is missing recommend you to please install the same.

Instructions to run the code when you are training and predicting.

1) Keep all files in one folder.
   trainer.py, data_transformer.py, model_runner.py, tox.ini, dataset_00_with_header.csv, holdout csv files.

2) Now run the below mentioned command to train the model.

   ```
   python model_runner.py tox_train.ini 1
   ```

   It will run for around 3 hours for the parameter set in tox_train.ini.

3) After completion of step 2, it will generate two pickle file – lgbm.pkl and catboost.pkl file. These file can now be used to load and predict for any set of holdout files that has same number of columns and transformation as dataset_00_with_header.csv.

4) Now change the file name in tox.ini with the filename of holdout file.
   run below mentioned for predicting from pickle files.

   python model_runner.py tox.ini 3

   If the holdout file contains target column 'y' then no need to change any other parameter in tox.ini. However, if the holdout file doesn't contain target column then change testcolhas_target variable to no.

   (I have tested the code with testcolhas_target='yes' but have not tested the code with testcolhas_target='no')

   ***CAUTION – holdout file and training file should have same columns.*

   The whole code set will output - submission file, transformed and cleaned dataset, model pickle files for lightgbm and catboost.

Instructions to run the code when you are only predicting.

1) Download pretrained pickle files from below mentioned link.
   https://www.kaggle.com/xooca1/lgbm-catboost-pretrained

   You will find two files: catboost.pkl and lgbm.pkl
2) Keep all files in one folder.
   trainer.py, data_transformer.py, model_runner.py, tox.ini, holdout csv files, catboost_list.pkl, lgbm_list.pkl, catboost.pkl and lgbm.pkl

3) Now change the file name in tox.ini with the filename of holdout file.
   run below mentioned for predicting from pickle files.

   python model_runner.py tox.ini 3

   If the holdout file contains target column 'y' then no need to change any other parameter in tox.ini. However, if the holdout file doesn't contain target column then change testcolhas_target variable to no.

   (I have tested the code with testcolhas_target='yes' but have not tested the code with testcolhas_target='no')

   ***CAUTION – holdout file and training file should have same columns.*

   The whole code set will output - submission file, transformed and cleaned dataset, model pickle files for lightgbm and catboost.

Submission file will contain following columns: (when ran with testcolhas_target='yes')

Id	id column
y	original target column
y_pred_lgbm	this is the predicted value from lighgbm model
y_pred_catboost	this is the predicted value from lighgbm model
y_pred_lgbm-y	this is difference between predicted lightgbm and actual
result_lgbm	this value is 0 if absolute error value is mode than 3 else 1 (for lightgbm)
y_pred_catboost-y	this is difference between predicted lightgbm and actual
result_catboost	this value is 0 if absolute error value is mode than 3 else 1 (for catboost)
y_pred_comb_max	this is the max value between lightgbm and catboost model.
y_pred_comb_max-y	this is difference between predicted and actual (for max of lighgbm and catboost)
result_comb_max	this value is 0 if absolute error value is mode than 3 else 1 (for max of lighgbm and catboost)
y_pred_comb_min	this is the min value between lightgbm and catboost model.
y_pred_comb_min-y	this is difference between predicted and actual (for min of lighgbm and catboost)
result_comb_min	this value is 0 if absolute error value is mode than 3 else 1 (for min of lighgbm and catboost)
y_pred_comb_mean	this is the mean value between lightgbm and catboost model.
y_pred_comb_mean-y	this is difference between predicted and actual (for mean of lighgbm and catboost)
result_comb_mean	this value is 0 if absolute error value is mode than 3 else 1 (for mean of lighgbm and catboost)
y_pred_comb_median	this is the median value between lightgbm and catboost model.
y_pred_comb_median-y	this is difference between predicted and actual (for median of lighgbm and catboost)
result_comb_median	this value is 0 if absolute error value is mode than 3 else 1 (for median of lighgbm and catboost)

_**if you are running with option 3 i.e. step 4 in instructions with testcolhas_target = 'no' then above yellow marked columns won't be generated._

Below is the accuracy of the pretrained model considering absolute error as 3:

The result is based on 20 percent of the train data:
{'result_lgbm': 0.1586, 'result_catboost': 0.12245, 'result_comb_max': 0.14375, 'result_comb_median': 0.14875, 'result_comb_mean': 0.14875, 'result_comb_min': 0.1373}

I found that model using lighgbm was having best accuracy compared to catboost. Also the combined result of the model also didn't beat lightgbm. I also tried the same using deep learning(pytorch) but that also didn't fetched good result.

Methodology:

1) There were many columns which was having lots of null values. First I removed crossed a threshold of 0.4. null_column_reject_threshold is the parameter in tox_train.ini that is being referred.
2) Then I imputed the rest of the columns using model imputers. I used random forest model imputer. I thought its best to give an algorithm control to impute values rather that putting mean or median values against them. I used separate model imputer for categorical and continuous values. Here those columns whose unique value is less than 20 is being treated

as categorical. This value can be changed by modifying cat_threshold in modifying tox_train.ini.

3) I tried using featurization , two column featurization , feature importance, one hot encoding high correlation with target, applying pca(refer data_tranformer.py) but they didn't worked. The rmse value was high when the model was running with these data transformation.
So I applied the most simplest transformation as below:

```
model_null_impute_cat_rf
model_null_impute_notcat_rf
retail_reject_cols
standardize_simple_auto
remove_collinear
convertdatatypes
```

4) I then ran the model through lightgbm and catboost model. And found that lightgbm was giving best model .