1. Data Preparation

a. Create three folders **train_fold**, **test_fold**, and **Skip_Data**, and put the decompressed train CSV files, test CSV files, and song feature files there
b. Create a new folder **test_pred** for storing the processed test files
c. Import the required modules and libraries; install them if needed
d. Specify the paths to the skip data, train data, test data, and test prehistory data
e. Double check the names for file1, file2 in the function “process_song_features”
f. Might need to change the path names for the parquet files being generated in the function “process_test_data”
g. Download the “vectors_150.txt” file from https://drive.google.com/file/d/1LZFI_UpPdfHagLvatmFp8qt5T7SGRj3P/view?usp=sharing and store it under the correct folder before running the function “process_glove_data”

2. Model Fitting

a. Create a new folder **Data** for storing the trained model
b. Import the required modules and libraries; install them if needed
c. Specify the skip data, and train data file paths [I have used only one of the train CSV files since my laptop could not support more; we can use “glob.glob(../train_fold/log_0_*_000000000000.csv)” instead to process >1 train files]
d. Might need to change the section between lines 91 and 127 accordingly
e. I have used a simple model to train due to a lack of computation power. The original model parameters used by the authors are commented between lines 533 and 542.

3. Test File Prediction

a. Import the required modules and libraries; install them if needed
b. Specify the skip data, test data file paths
c. Specify/change the path to save the best model in line 385
d. Adjust the item indices between lines 387 and 434

4. Submission Generation

a. Import the required modules and libraries; install them if needed
b. Specify the skip data, test data file paths
c. This code generates the skip prediction in the format required by the challenge, and stores it in the file named “submission_0108_new.txt” i.e. each represents the skip prediction of the tracks in that particular session

5. Probability

a. Import the required modules and libraries; install them if needed
b. Specify the skip data, test data file paths
c. Check/modify the paths on lines 29 and 58

d. This code takes the probabilities from the model and adds them as a column to the dataset, saved as “new_data”

6. Prepare New Data Counterfactual

a. Specify the test data path, the track id of the track we need a counterfactual for, the session length, the session position of the track, and the session id
b. Check/modify the paths on lines 8 and 27
c. This code creates a new data set with entries as all of the unique track ids except the one to be replaced keeping the other parameters the same

7. Counterfactual Prediction

a. Import the required modules and libraries; install them if needed
b. Specify the skip data, test data file, and counterfactual data paths
c. Specify/change the path to save the best model in line 386
d. Adjust the item indices between lines 388 and 472
e. This code treats the new counterfactual file as a new test data file and predicts whether the tracks will be skipped or not