Readme

General instruction:

1. For compiling the MapReduce code
    1. The MapReduce file is always named as xxxx.java, and the mapper file is always named as xxxx_mapper.java, while the reducer file is always named as xxxx_reducer.java
    2. javac -classpath `yarn classpath` -d . xxxx.java xxxx_mapper.java xxxx_reducer.java
    3. jar -cvf xxxx.jar *.class
    4. hadoop jar xxxx.jar xxxx input_file_directory output_file_directory

To run the code by steps:

1. First we use the files in csv_files3 and profiling.jar to obtain the 4 files in which the word "county" was removed and the FMR of five types of rooms are extracted.
    1. First put all of the files in the csv_files3 folder into your hdfs storage system.
    2. Run the following command in dumbo, in the directory where your MapReduce code are being stored.
    3. javac -classpath `yarn classpath` -d . profiling.java profiling_mapper.java profiling_reducer.java
    4. jar -cvf profiling.jar *.class
    5. hadoop jar profiling.jar profiling your_hdfs_directory/FY2021_50_County.csv your_hdfs_output_folder
    6. Repeat the above for three times for different input files: FY2018_50_County_rev.csv, FY2019_50_County_rev.csv, FY2020_50_County_rev.csv. Remember to use a new output directory for each time.
    7. Use hdfs dfs -getmerge output_directory your_text.txt to retrieve those four files to your local directory, and merge them together to form a single txt or csv file, lets call it the four_years.txt.
2. Secondly we are going to process the covid by CovidByCountyOriginal.txt using combine.jar.
    1. javac -classpath `yarn classpath` -d . combine.java combine_mapper.java combine_reducer.java
    2. jar -cvf combine.jar *.class
    3. hadoop jar combine.jar combine your_hdfs_directory/CovidByCountyOriginal.txt your_hdfs_output_folder
    4. Use hdfs dfs -getmerge output_directory your_text.txt to retrieve the output, lets call it the 10-01ByCounty.txt.
3. Thirdly, use the mapping.jar to process the 10-01ByCounty.txt and four_years.txt.
    1. javac -classpath `yarn classpath` -d . mapping.java mapping_mapper.java mapping_reducer.java
    2. jar -cvf mapping.jar *.class
    3. hadoop jar mapping.jar mapping your_hdfs_directory/CovidByCountyOriginal.txt your_hdfs_output_folder
    4. Use hdfs dfs -getmerge output_directory your_text.txt to retrieve the output, lets call it the mapped_file.txt.
4. Fourthly, use the evaluate.jar to process the mapped_file.txt
    1. javac -classpath `yarn classpath` -d . evaluate.java evaluate_mapper.java evaluate_reducer.java
    2. jar -cvf evaluate.jar *.class
    3. hadoop jar evaluate.jar evaluate your_hdfs_directory/mapped_file.txt your_hdfs_output_folder
    4. Use hdfs dfs -getmerge output_directory your_text.txt to retrieve the output, lets call it the evaluated_file.txt.
5. Fifthly, use the predict.jar to process the mapped_file.txt.

   1. javac -classpath `yarn classpath` -d . predict.java predict_mapper.java predict_reducer.java
   2. jar -cvf predict.jar *.class
   3. hadoop jar predict.jar predict your_hdfs_directory/mapped_file.txt your_hdfs_output_folder
   4. Use hdfs dfs -getmerge output_directory your_text.txt to retrieve the output, lets call it the predicted_file.txt.

6. Sixthly, use the evaluated_file.txt and Hive to sort out the 50 top counties whose rental prices were negatively impacted most.
   1. create external table e2 (data1 string, data2 string, data3 string, data4 int,data5 int,data6 int,data7 int,infection int, average_chge int) row format delimited fields terminated by "," location "your_hdfs_directory_of_evalutede_file_txt";
   2. select * from e2 order by average_chge desc limit 50;
   3. select count(*) from e2 where data4=-1;

By far, you have successfully run all the code of the county FMR price and Covid-19 impact. If you have any problem please email me xw2447@nyu.edu, so that I can give you more detailed instruction if needed.