1. **DATA SOURCES**

We have four Data Sources:

1. NYPD Motor Vehicle Collisions :

* LINK : <https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95>
* File name in directory: /datasets/NYPD\_Motor\_Vehicle\_Collisions.csv

1. Parking Violations Issued :

* LINK : https://data.cityofnewyork.us/City-Government/Parking-Violations-Issued-Fiscal-Year-2017/2bnn-yakx
* File name in directory: /datasets/Parking\_Violations\_Issued.csv

1. Moving Violations Summonses :

* LINK : <https://catalog.data.gov/dataset/traffic-tickets-issued-beginning-2008>
* File name in directory : /datasets/Traffic\_Tickets\_Issued.csv

1. Weather data in New York City - 2016 :

* LINK : <https://www.kaggle.com/mathijs/weather-data-in-new-york-city-2016>
* File name in directory : /datasets/weather\_data\_nyc\_centralpark\_2016.csv

1. **DATA INGESTION**

We put the data files from the system’s memory to Dumbo local File system using tools like FUGU and FileZilla.  
Next, the data was injected into Hadoop HDFS using commands mentioned in the file,

‘/data\_ingest/ingest.sql’

1. **ETL**

The ETL related codes (including java files, class files and jar files) can be found in the /etl\_code folder. The files can be executed using the following commands:

*PARKING VIOLATIONS*

* javac -classpath `yarn classpath` -d . ParkingViolationETLMapper.java -- to compile the java file (class file already provided)
* javac -classpath `yarn classpath`:. -d . ParkingViolationETL.java -- to compile the java file (class file already provided)
* jar -cvf parkViolation.jar ParkingViolation\*.class -- to create the jar file (file already provided)
* hadoop jar parkViolation.jar ParkingViolationETL project/Parking\_Violations\_Issued.csv project/Parking\_ETL\_final\_output -- to execute the parking violation ETL MapReduce job

*MOTOR COLLISIONS*

* javac -classpath `yarn classpath` -d . MotorCollisionETL/MapperClass.java -- to compile the java file (class file already provided)
* javac -classpath `yarn classpath`:. -d . MotorCollisionETL/Cleanetl.java -- to compile the java file (class file already provided)
* jar -cvf Cleanetl.jar MotorCollisionETL/\*.class -- to create the jar file (file already provided)
* hadoop jar Cleanetl.jar Cleanetl /project/NYPD\_Motor\_Vehicle\_Collisions.csv /project/Motor\_Collisions\_ETL\_final\_output

*TRAFFIC TICKETS SUMMONSES*

* javac -classpath `yarn classpath` -d . TrafficTicketsIssuedETLMapper.java -- to compile the java file (class file already provided)
* javac -classpath `yarn classpath`:. -d . TrafficTicketsIssuedETL.java -- to compile the java file (class file already provided)
* jar -cvf TrafficTicketsIssuedETL.jar TrafficTicketsIssued\*.class -- to create the jar file (file already provided)
* hadoop jar TrafficTicketsIssuedETL.jar TrafficTicketsIssuedETL project/Traffic\_Tickets\_Issued.csv project/Tickets\_ETL\_final\_output -- to execute the traffic tickets ETL code

*WEATHER DATA IN NYC*

The data did not need ETL, profiling has been done in the next step.

1. **PROFILING**

The codes related to profiling (including java files, class files and jar files) can be found in the /profiling\_code folder. The files can be executed using the following commands:

*PARKING VIOLATIONS*

* javac -classpath `yarn classpath` -d . ParkingViolationProfilingMapper.java -- to compile the java file (class file already provided)
* javac -classpath `yarn classpath`:. -d . ParkingViolationProfiling.java -- to compile the java file (class file already provided)
* jar -cvf parkingProfile.jar ParkingViolationProf\*.class -- to create the jar file (file already provided)
* hadoop jar parkingProfile.jar ParkingViolationProfiling project/Parking\_Profile\_final\_output -- to execute the parking violation ETL code

*MOTOR COLLISIONS*

* The cleaned data did not require profiling.

*TRAFFIC TICKETS SUMMONSES*

* javac -classpath `yarn classpath` -d . TrafficTicketsIssuedProfilingMapper.java -- to compile the java file (class file already provided)
* javac -classpath `yarn classpath`:. -d . TrafficTicketsIssuedProfiling.java -- to compile the java file (class file already provided)
* jar -cvf TrafficTicketsIssuedProfiling.jar TrafficTickets\*.class -- to create the jar file (file already provided)
* hadoop jar TrafficTicketsIssuedProfiling.jar TrafficTicketsIssuedProfiling project/Tickets\_ETL\_final\_output/part-r-00000 project/Tickets\_Profile\_output -- to execute the traffic tickets Profiling code

ITERATION 2:

* javac -classpath `yarn classpath` -d . FinalTrafficTicketsIssuedMapper.java

-- to compile the java file (class file already provided)

* javac -classpath `yarn classpath`:. -d . FinalTrafficTicketsIssued.java

-- to compile the java file (class file already provided)

* jar -cvf FinalTrafficTicketsIssued.jar FinalTrafficTickets\*.class

-- to create the jar file (file already provided)

* hadoop jar FinalTrafficTicketsIssuedProfiling.jar FinalTrafficTicketsIssued project/Tickets\_Profile\_output/part-r-00000 project/Tickets\_output\_2

-- to execute the traffic tickets Profiling code

FINAL :

* javac -classpath `yarn classpath` -d . TrafficTicketsProfileFinalMapper.java

-- to compile the java file (class file already provided)

* javac -classpath `yarn classpath`:. -d . TrafficTicketsProfileFinal.java

-- to compile the java file (class file already provided)

* jar -cvf TrafficTicketsProfileFinal.jar TrafficTicketsProfile\*.class

-- to create the jar file (file already provided)

* hadoop jar TrafficTicketsProfileFinal.jar TrafficTicketsProfileFinal project/Tickets\_output\_2/part-r-00000 project/Tickets\_Profile\_final\_output

-- to execute the traffic tickets Profiling code

*WEATHER DATA IN NYC*

* javac -classpath `yarn classpath` -d . WeatherProfileMapper.java -- to compile the java file (class file already provided)
* javac -classpath `yarn classpath`:. -d . WeatherProfile.java -- to compile the java file (class file already provided)
* jar -cvf weatherProfile.jar WeatherProfile\*.class -- to create the jar file (file already provided)
* hadoop jar weatherProfile.jar WeatherProfile project/weather\_data\_nyc\_centralpark\_2016.csv project/Weather\_Profile\_final\_output -- to execute the weather profile MapReduce job

1. **EXTRACTING OUTPUT FILES**

The output files from ETL and Profiling stage needed to be put into the project packet, hence were extracted from HDFS using commands mentioned in ‘/data\_extraction/extraction.sql’ file.  
After having run these commands, the files can be brought into the local system using tools like FUGU or FileZilla.   
Also, the output files have been uploaded in the respective folders,   
E.g. output file for Parking Violations ETL is stored as -- /data\_ETL/parkingETLOutput ,

output file for Parking Violations Profile is stored as -- /data\_ETL/parkingProfilingOutput , and so on.

1. **ANALYTICS**

* ANALYTIC 1: Correlation between petty offenses (parking violations and petty moving violations) and motor collisions

The code and outputs related to this file can be found in the folder “analytic/analytic1/code\_an1.sql” and “analytic/analytic1/output” files respectively

The screenshots are present in the “/screenshots/analytic\ 1” folder.

* ANALYTIC 2: Spotting a relation between changing weather and number of motor collisions

The code related to this file can be found in the folder “analytic/analytic3/code\_an3.sql”

* ANALYTIC 3: Identification of “hot-spots”, i.e. locations where collisions are a frequent phenomenon

The code and output related to this file can be found in the folder “analytic/analytic3/code\_an3.sql” and “analytic/analytic3/output” files respectively. The visualization of the output is present in “analytic/analytic3/visualization/”

* ANALYTIC 4: Spotting a relation between the days of week and number of motor collisions

The code and outputs related to this file can be found in the folder “analytic/analytic3/” and “analytic/analytic4/output” files respectively. The commands to run the code are mentioned in “Commands to Run the analytic.txt” file. The visualization of the output is present in “analytic/analytic4/visualization/”

* ANALYTIC 5: Spotting a relation between offenses done by age groups. The code and outputs related to this file can be found in the folder “analytic/analytic5/code\_an5.sql” and “analytic/analytic5/output” files respectively. The screenshots are present in the “/screenshots/analytic\ 5” folder.