**BDAD PROJECT DATA INGESTION REPORT**

**How Safe NYC really is?**

**Name**: Sanjeet Vijaya Shetty

**NetID**: ss14431

**Dataset 1**: **NYPD Complaints**

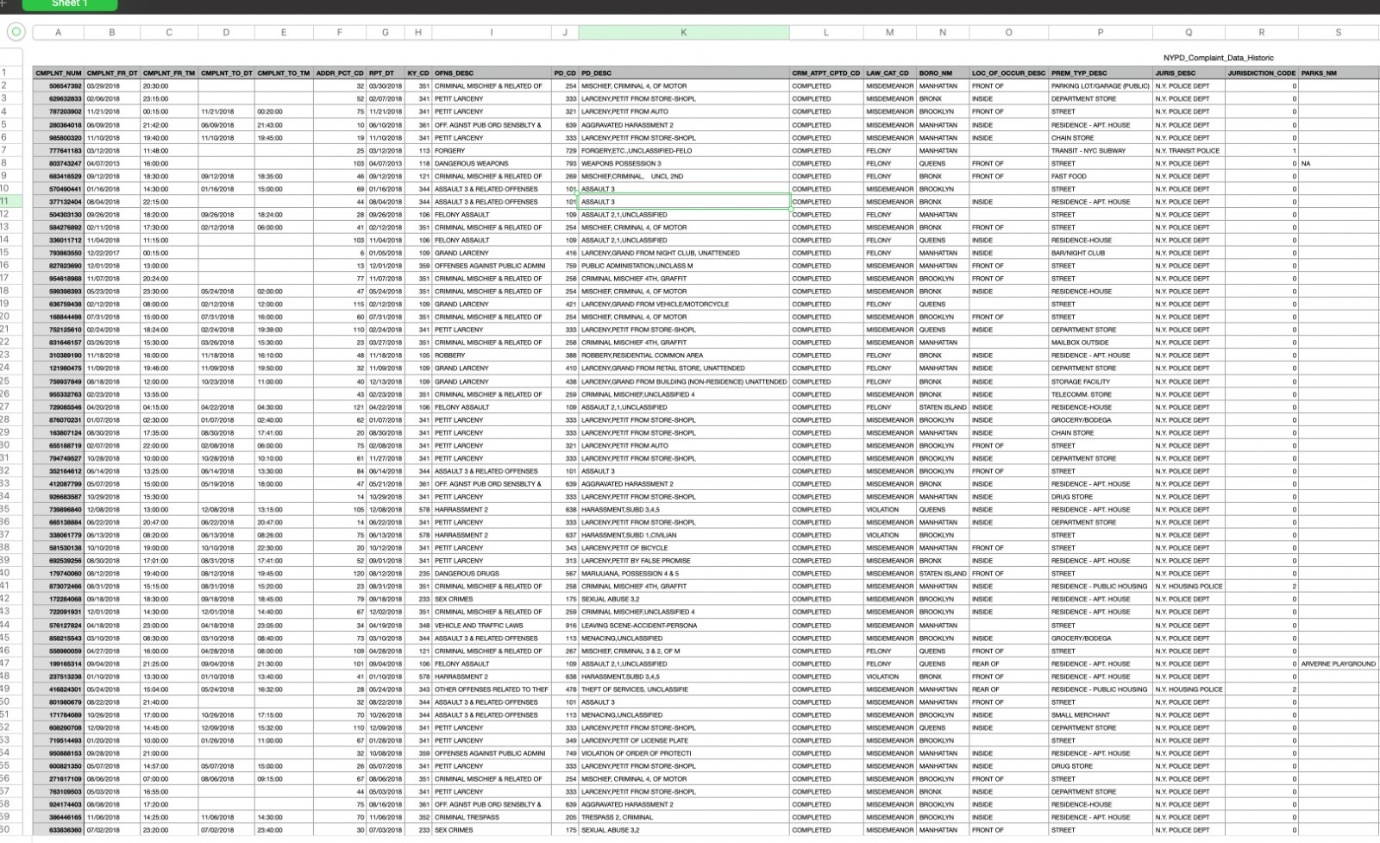
The NYPD also owns the dataset that follows. This dataset comprises every legitimate felony, misdemeanor, and violation crime that the New York City Police Department (NYPD) received reports of from 2006 through the end of the previous year. It was made public on November 16, 2016, and it is updated yearly.

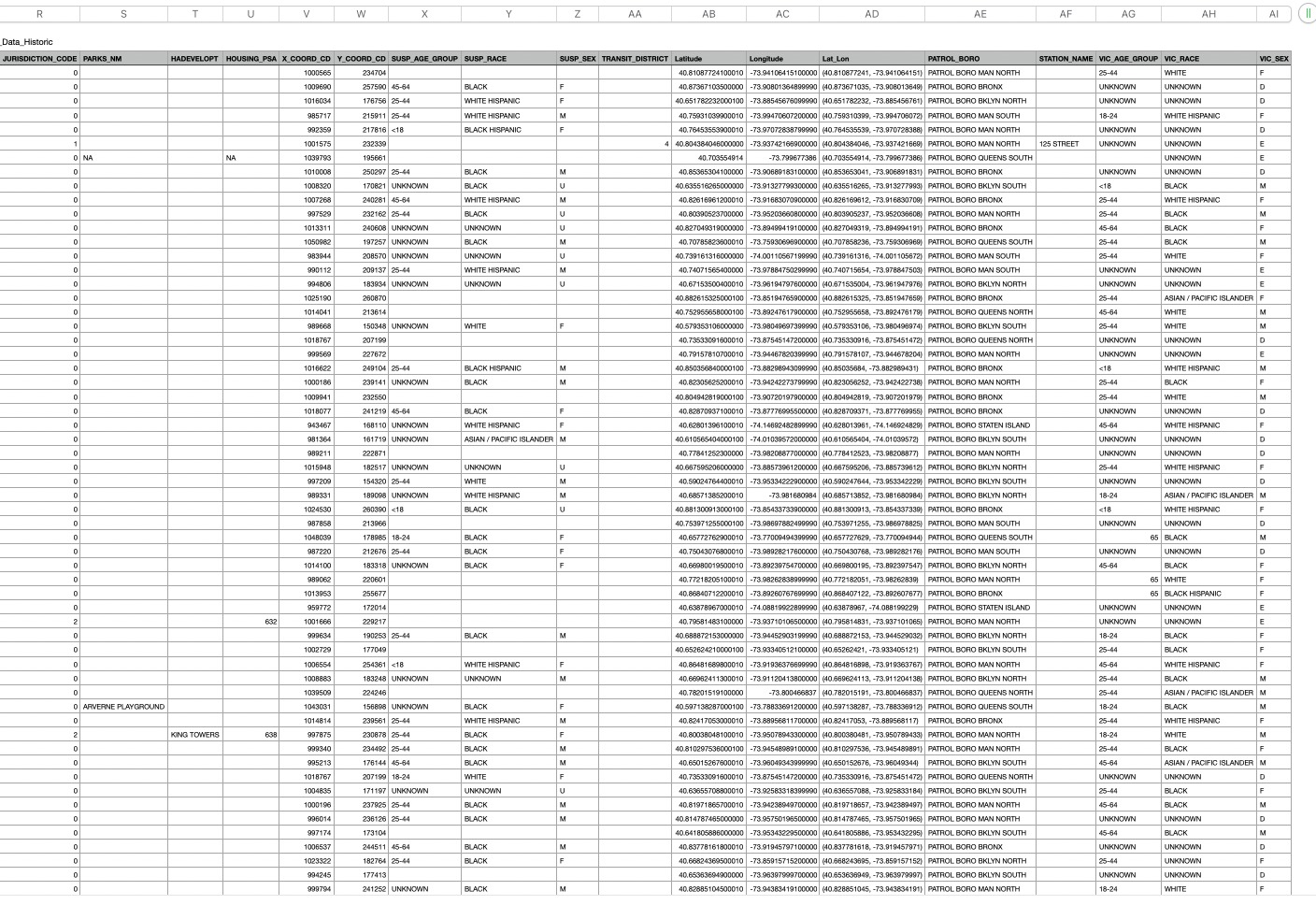
This dataset includes all valid felony, misdemeanor, and violation crimes reported to the New York City Police Department (NYPD) from 2006 to the end of last year.

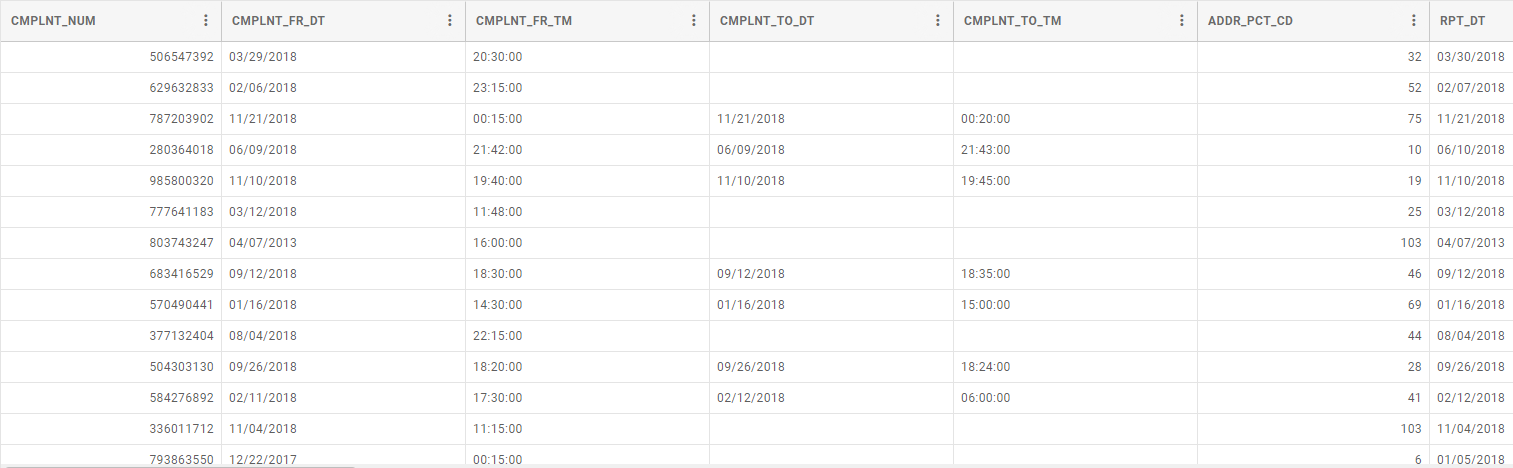
**Link for the dataset**: <https://data.cityofnewyork.us/Public-Safety/NYPD-Complaint-Data-Historic/qgea-i56i>

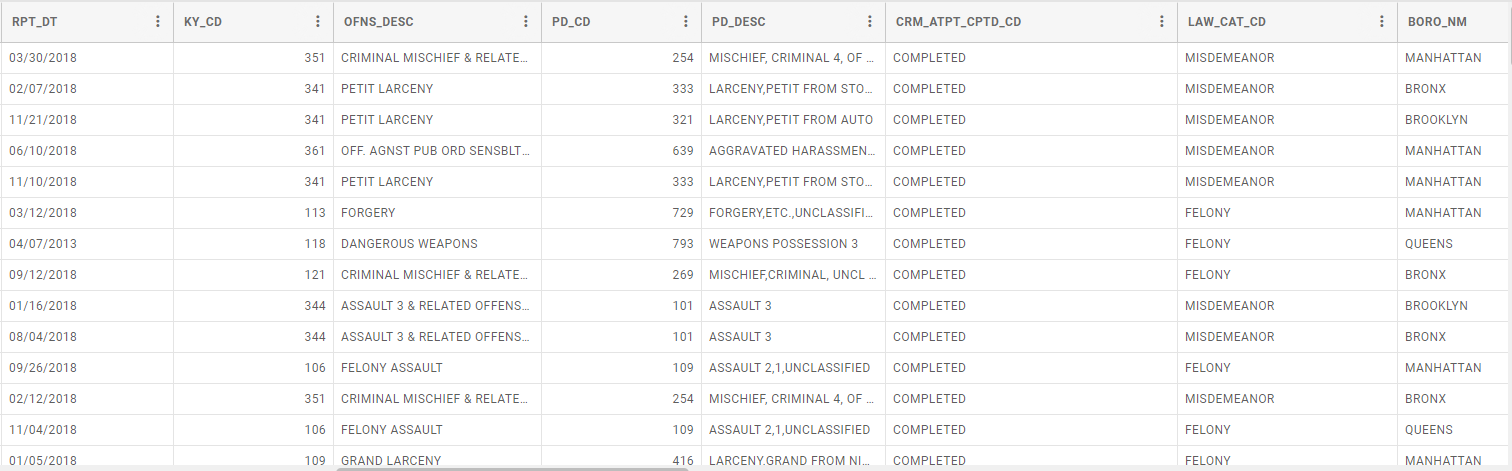
**File Size**: 2.51 GB

**Preview of Dataset**:







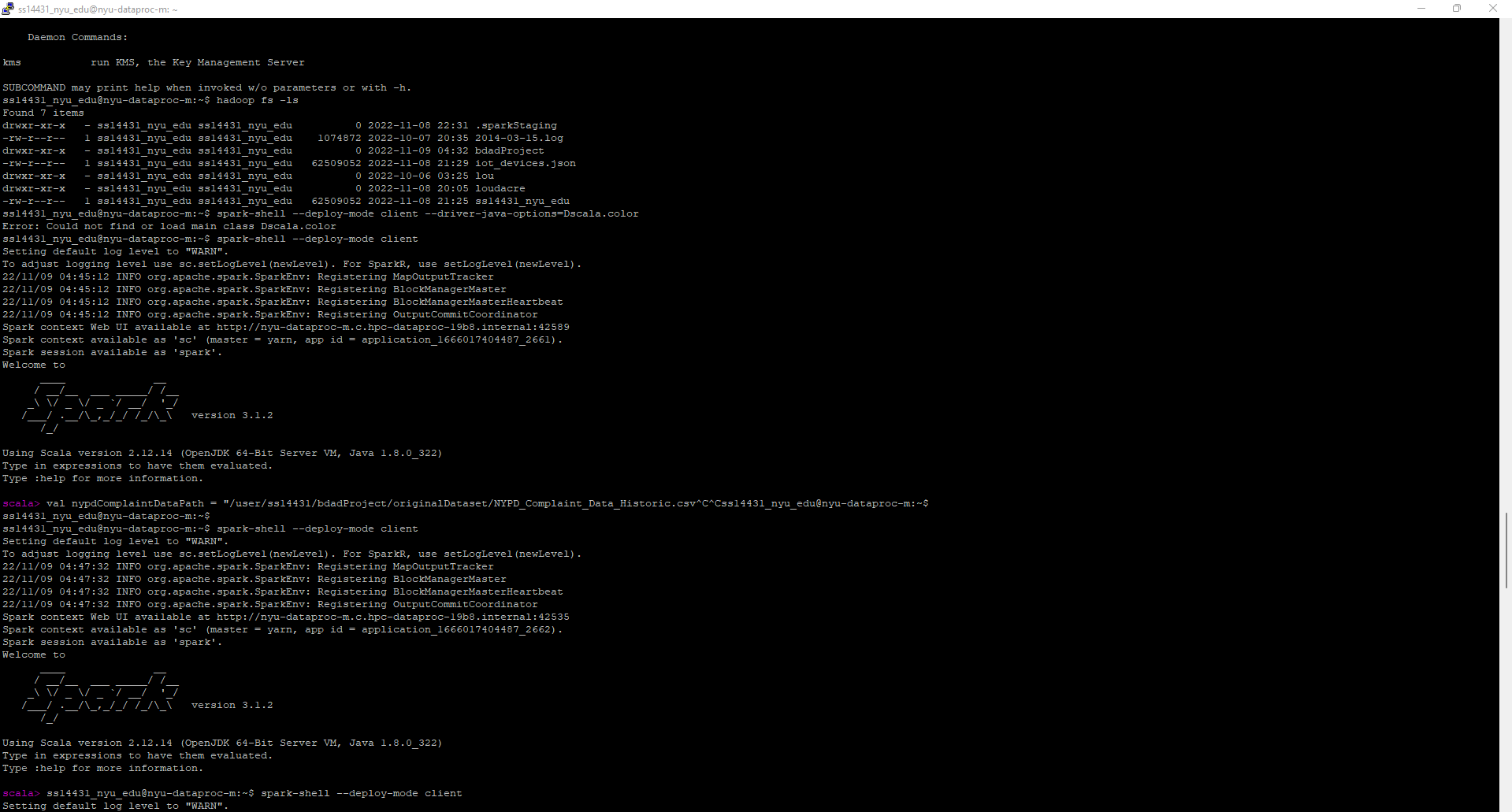


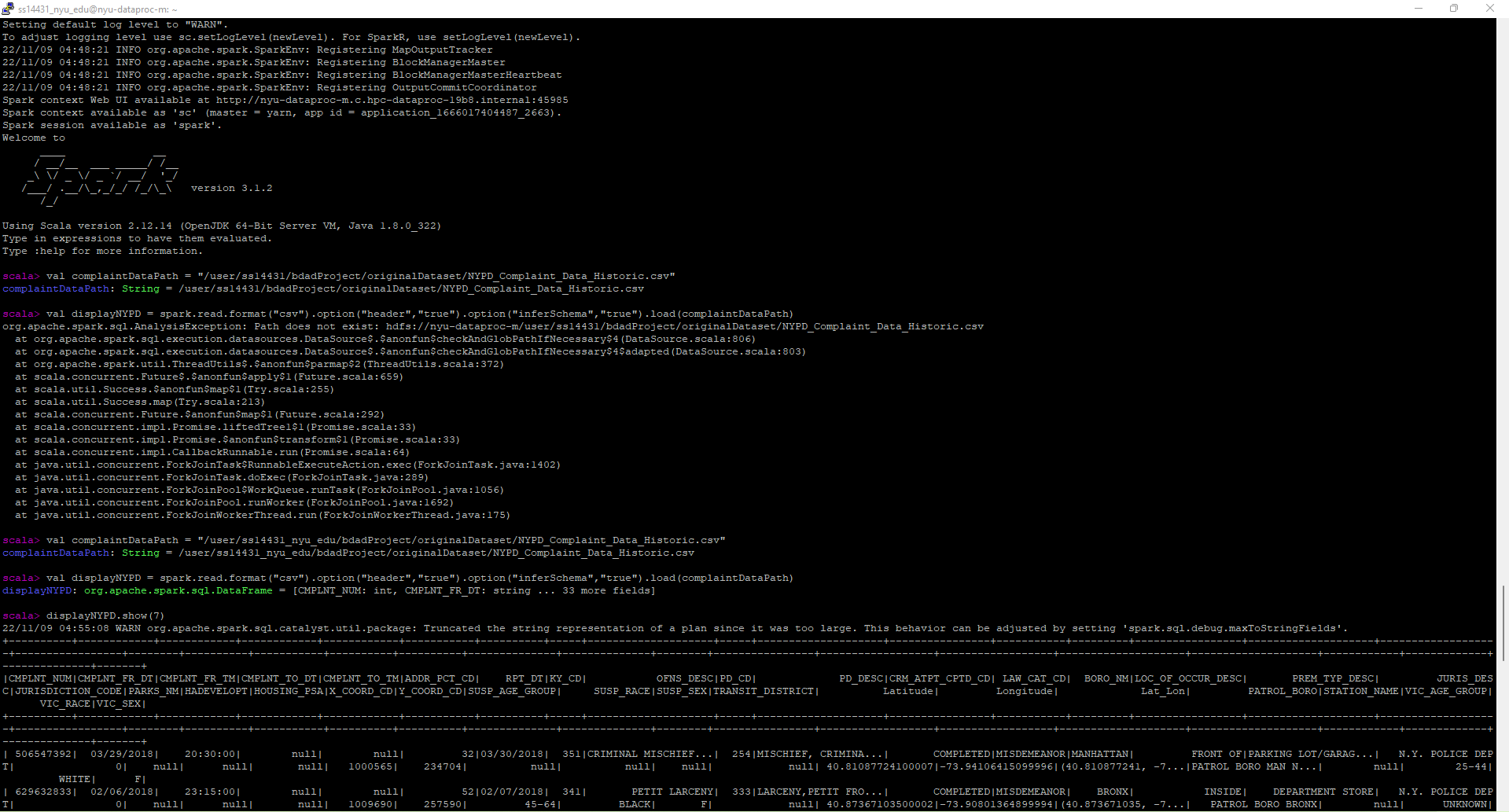
One of the initial steps for data pre-processing was deciding the columns that were required for the project. I chose the following columns to move forward with the project:

* COMPLAINT-NUM
* COMPLAINT DATE
* NYPD PRETINCT
* REPORT DATE
* DIGIT OFFENSE CODE
* OFFENSE DESC
* LEVEL OF OFFENSE
* BOROUGH NAME
* LOCATION OF OCCURRENCE
* JURISDICTION CODE
* NAME OF HOUSING
* HOUSING\_CODE
* SUSP\_AGE\_GROUP
* SUSP\_RACE
* SUSP\_SEX
* TRANSIT\_DISTRICT
* PATROL\_BORO
* STATION\_NAME
* VIC\_AGE\_GROUP
* VIC\_SEX

The columns listed above are necessary in order to do a real world analysis of identifying the complaints in NYC ,Types of Complaints ,Location where complaints are the most,the frequency of complaints and deriving some meaningful relationship which connects the above columns.

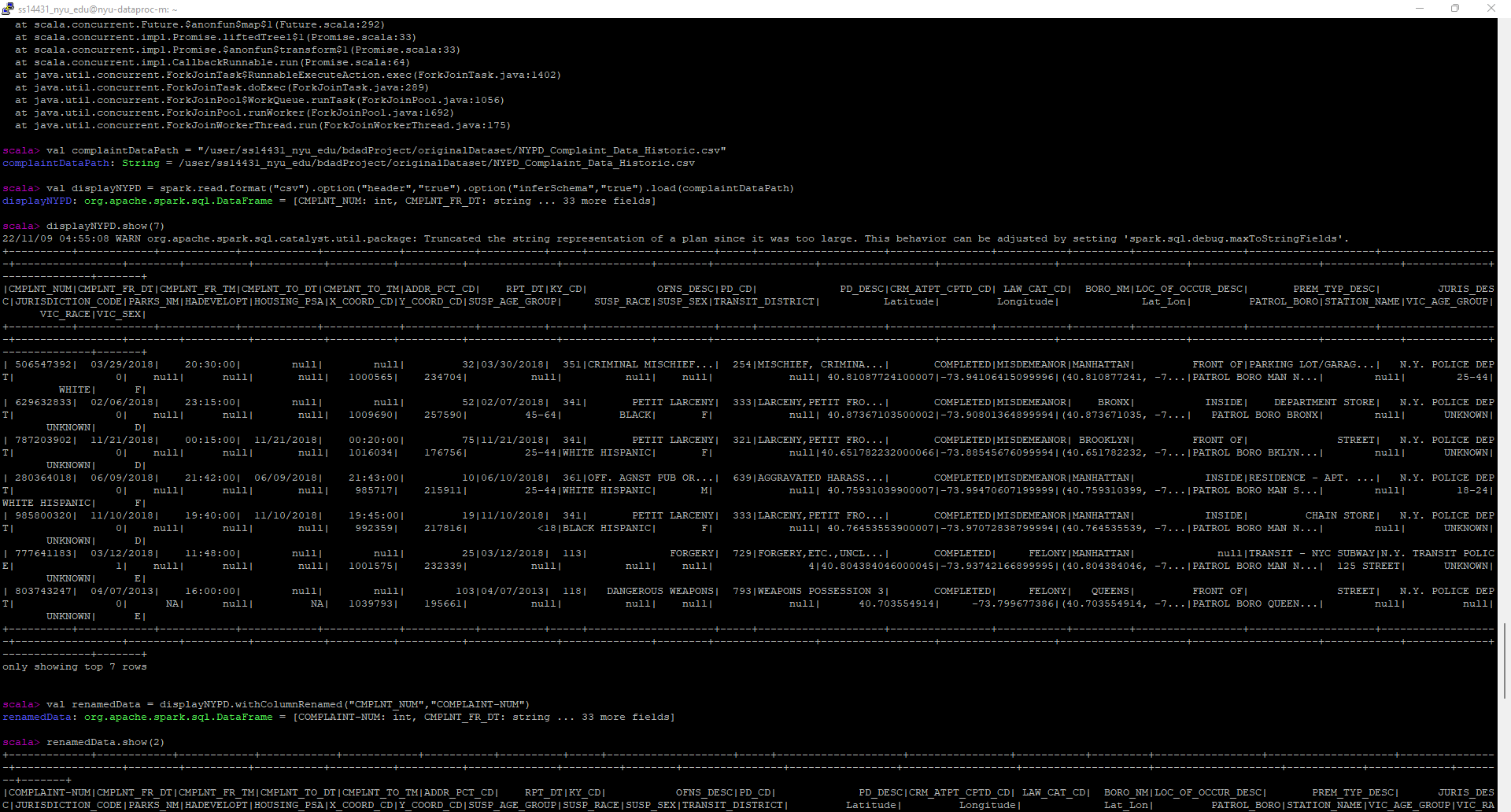
**Screenshots**:





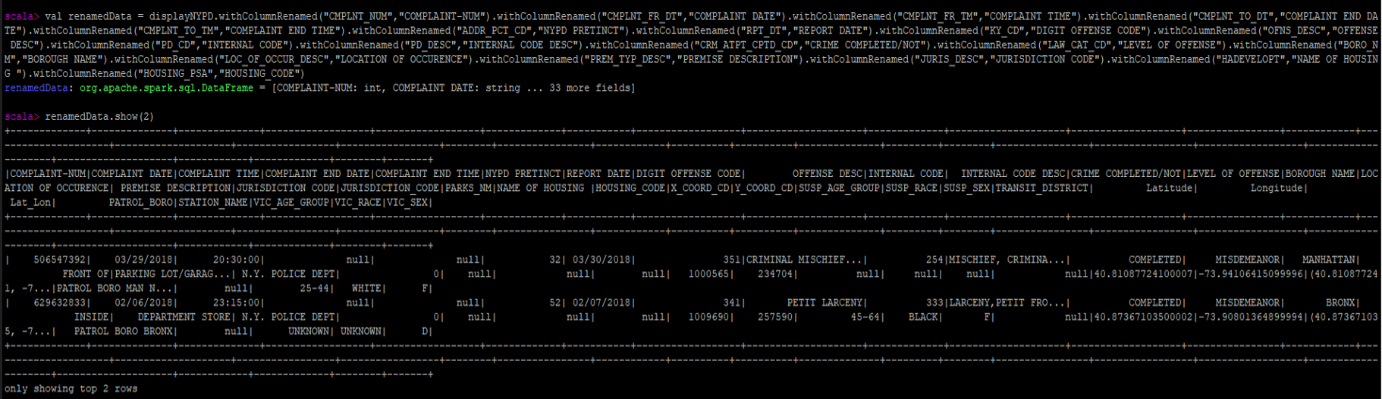
Initially, the path of the dataset is stored in a variable in Scala.(complaint Data Path)

We read the dataset which is of the format .csv in Scala and display the top 10 rows to crosscheck if the dataset is loaded properly.



**Data Processing**:

**Step 1**: Renaming Columns in order to make them readable in Normal form

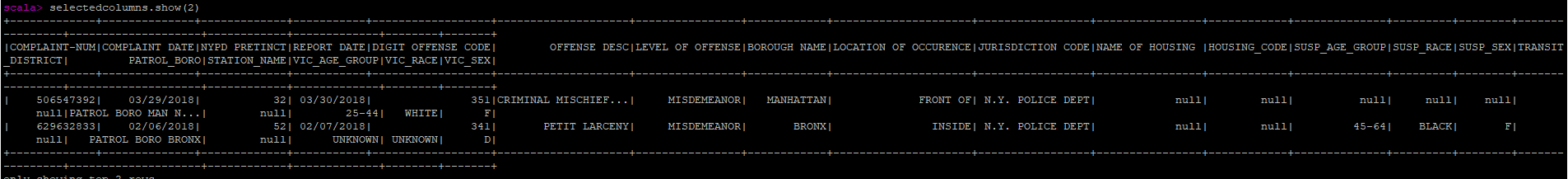


In ETL process,dropping data columns that are not required.The following command in Spark does the Job:



The above command selects only those columns that are mentioned,the other columns are dropped.

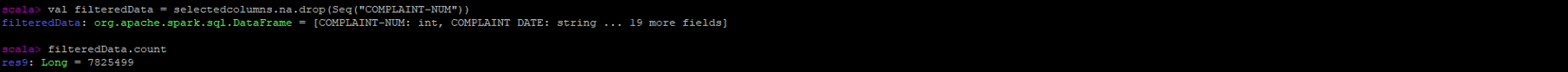
Dataframe after dropping unnecessary columns:



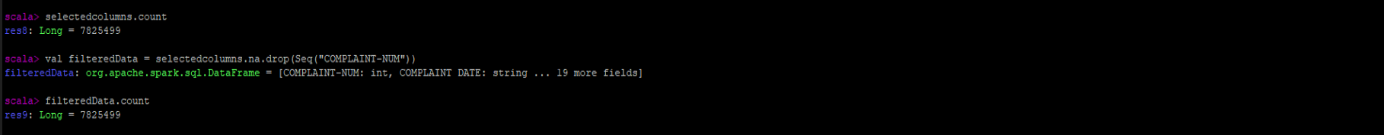
**Step 2**:

In the next step in data processing is dropping rows that have all null values i.e we need to drop redundant rows from our dataset.

val filteredData = selectedcolumns.na.drop(Seq(“COMPLAINT-NUM”))



The original data from NYC Complaint Data contained 7825499 rows:

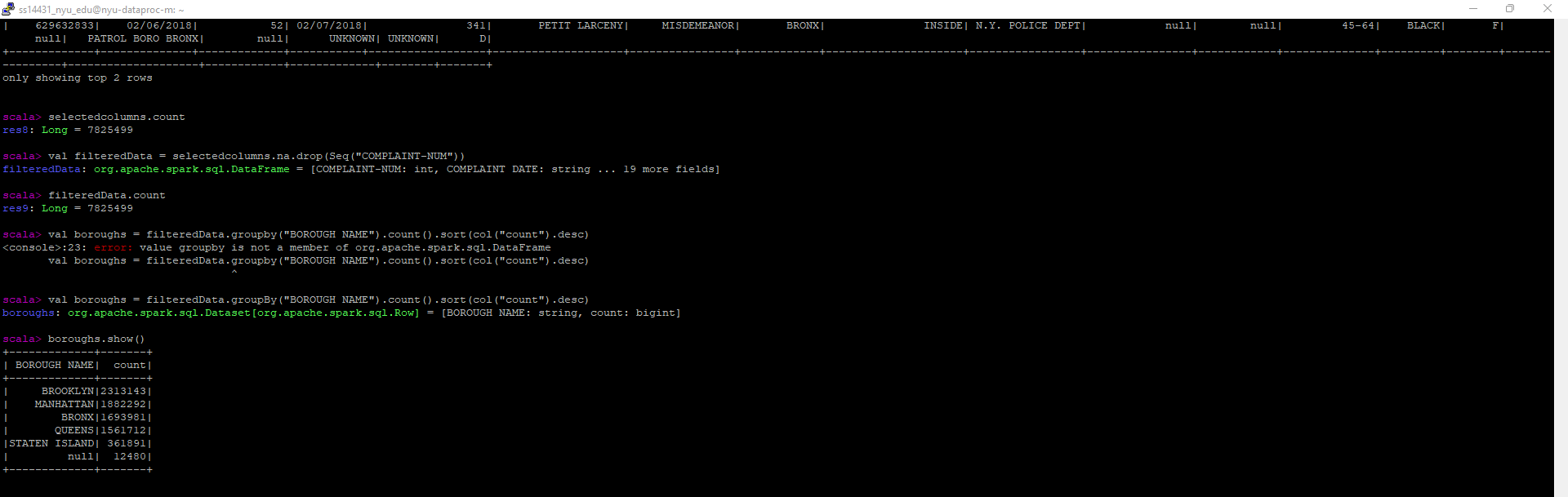


We can see from above command that none of the rows were dropped from the original 7825499 rows hence every row has a complaint number assigned to it.

**Valuable Insights from the Data**:

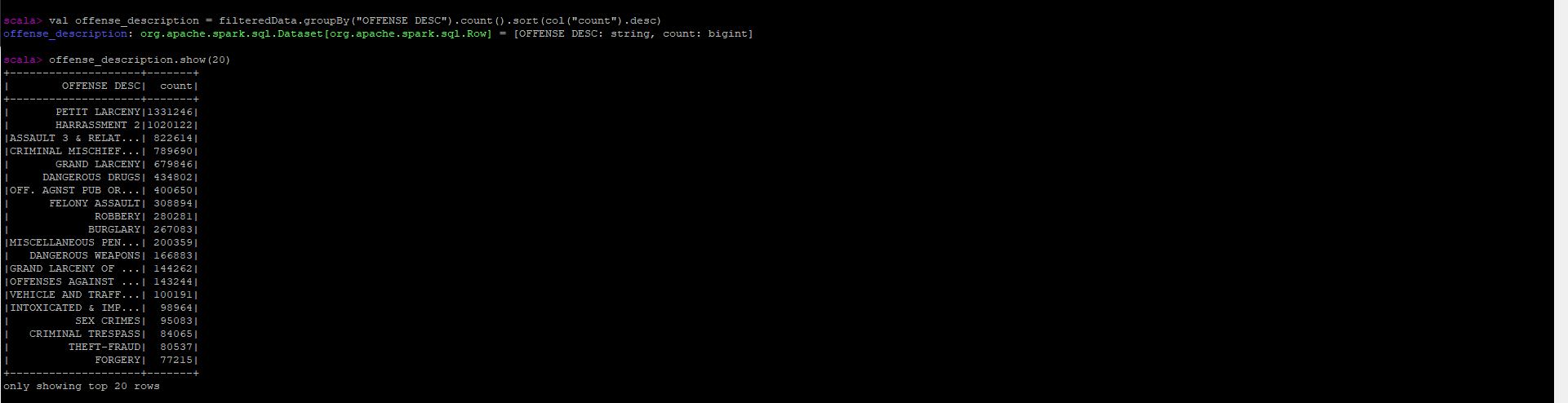
1)**Boroughs**- We are able to answer the question of **which borough** has the maximum number of complaints in NYC.

Used **groupBy() function** in spark for this purpose and displayed in descending order of count the number of complaints in the Boroughs



2)**Offense Description**-We are able to answer the question of **which offense** occurred the most in NYC across all Boroughs.

Used **groupBy() function** in spark for this purpose and displayed in descending order of count the number of offenses in the Boroughs



3)**Suspect Age Group**-We are able to answer the question of **which age group** is conducting the most offenses across the Boroughs.

Used **groupBy() function** in spark for this purpose and displayed in descending order of count the number of offenses in the Boroughs across Suspect Age Group.

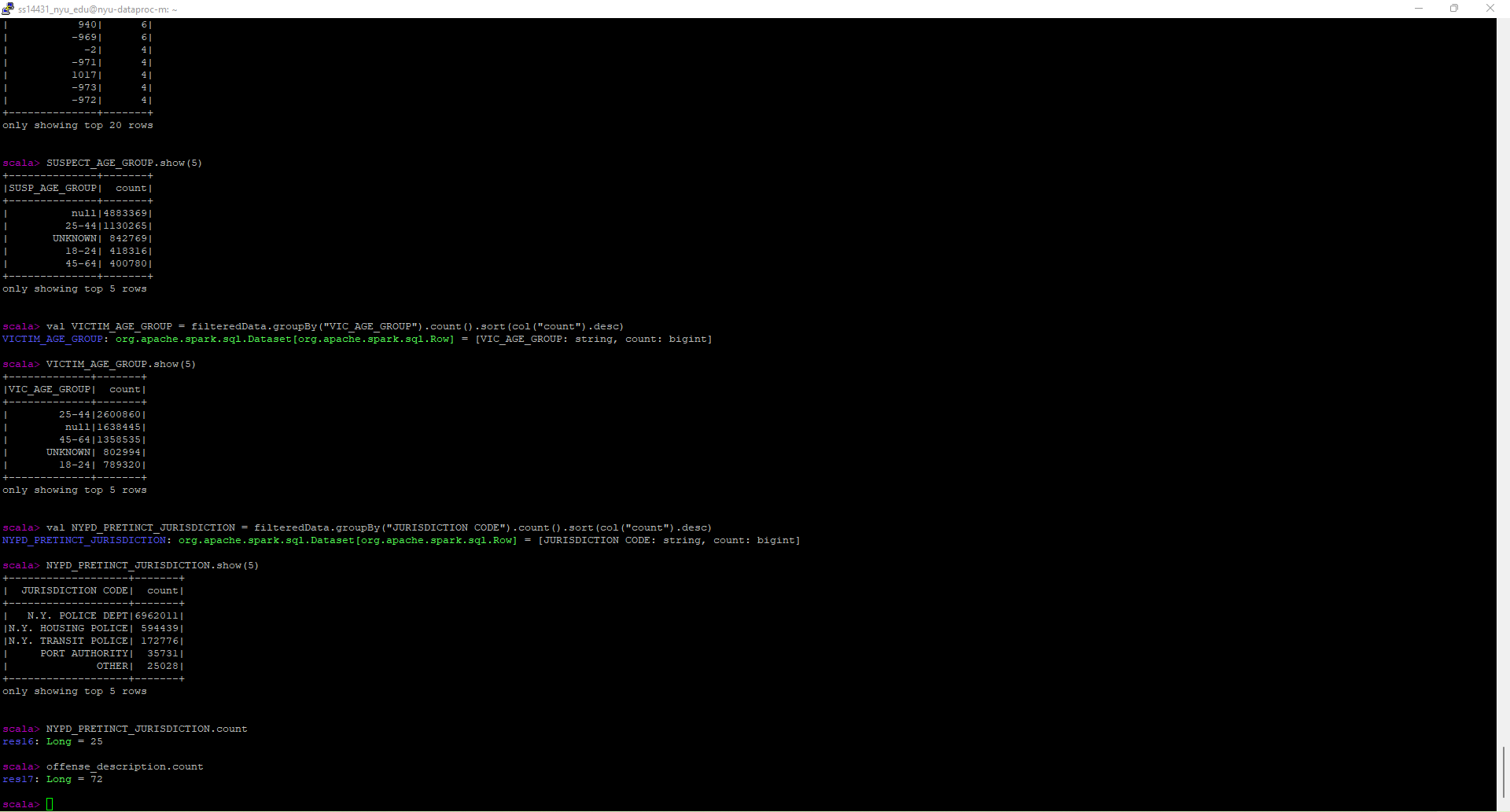


4) **Victim Age Group**-We are able to answer the question of **which age group** is most suspectible to offenses across the Boroughs.

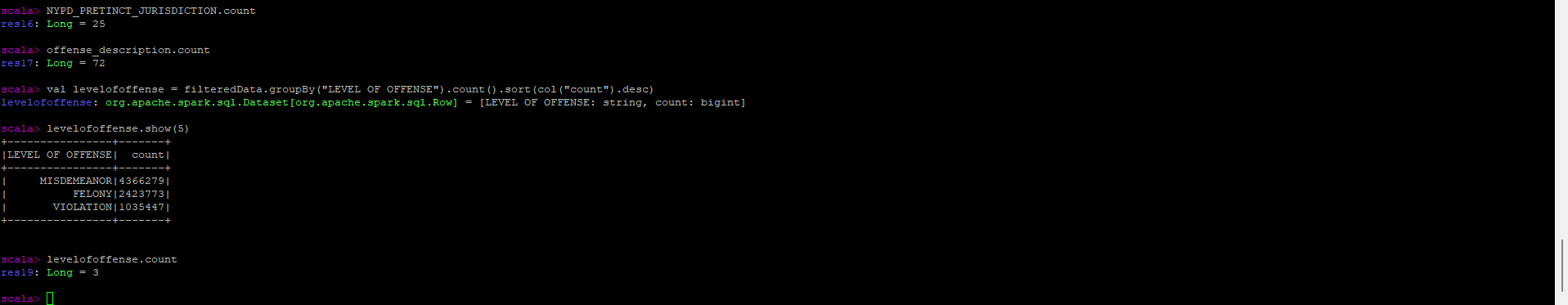
Used **groupBy() function** in spark for this purpose and displayed in descending order of count the number of offenses in the Boroughs across Victim Age Group.

5)**NYPD PRETINCT JURISDICTION** - We are able to answer the question of **which** **NYPD PRETINCT JURISDICTION** is handling the most complaints across the Boroughs.

Used **groupBy() function** in spark for this purpose and displayed in descending order of count which NYPD Pretinct Jurisdiction is handling most complaints .



6)**LEVEL OF OFFENSE**-We are able to answer the question of **which LEVEL OF OFFENSE** is present most across the Boroughs.

Used **groupBy() function** in spark for this purpose and displayed in descending order of count the number of offenses in the Boroughs across Victim Age Group

**Saved cleaned dataset in Hadoop for further use**:



**Overview of Scala Code used in the above Dataset**:

1)val complaintDataPath= "/user/ss14431\_nyu\_edu/bdadProject/orignalDataSet/NYPD\_Complaint\_Data\_Historic.csv"

2)val displayNYPD=spark.read.format("csv").option("inferSchema","true").load(complaintDataPath)

3)displayNYPD.show(7)

4)renamedData=displayNYPD.withColumnRenamed("CMPLNT\_NUM","COMPLAINT-NUM")

5)renamedData.show(2)

6) val selectedColumns=renamedData.select("COMPLAINT-NUM",...,"VIC\_SEX")

7)selectedcolumns.show(2)

8)val filteredData=selectedcolumns.na.drop(Seq("COMPLAINT-NUM"))

9)filteredData.count

10)selectedcolumns.count

11)val filteredData= selectedcolumns.na.drop(Seq("COMPLAINT-NUM"))

12)filteredData.count

13)val boroughs=filteredData.groupBy("BOROUGH NAME").count().sort(col("count").desc)

14)boroughs.show()

15)NYPD\_PRETINCT\_JURISDICTION.count

16)offense\_description.count

17)levelofoffense.count

18)val df=spark.read.format("csv").option("header","true").load("bdadProject/cleaneddataset/NYPDComplaint/NYPD\_cleaned.csv")

**Dataset 2**: **NYC 2010-2020 Census Data**

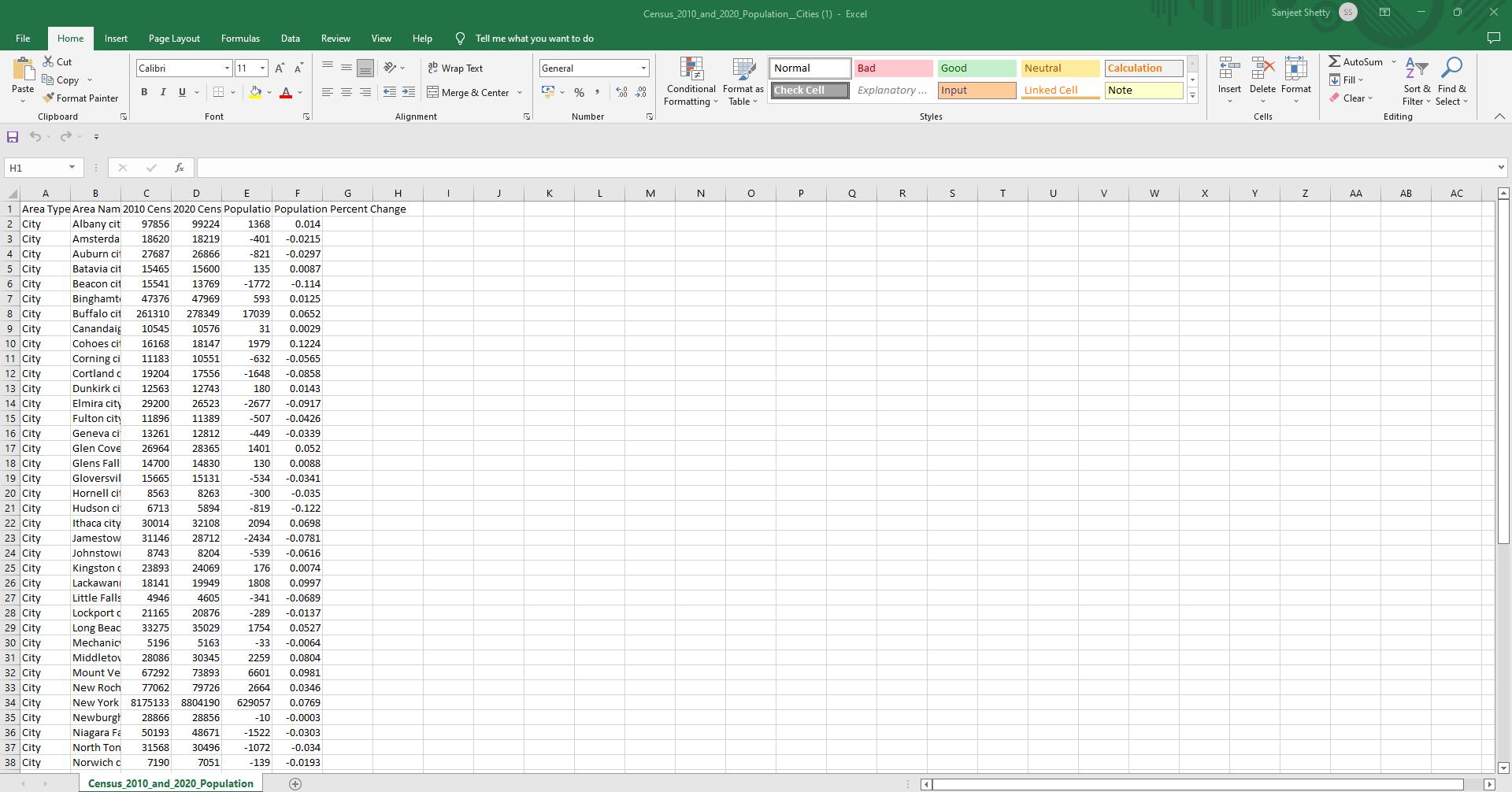
The NYC government owns the dataset that follows. Dataset does a comparison in the census statistics between the years 2010 and 2020. This dataset comprises different information about the basic housing and demographic information for different NYC neighborhoods. The updation of census data happens every 10 years.

**Link for the dataset**:

<https://www.nyc.gov/site/planning/https://catalog.data.gov/dataset/new-york-city-population-by-borough-1950-2040planning-level/nyc-population/2020-census.page>

**File size**: 2 MB

**Preview of Dataset**:



Here, we have 6 columns-

1)Area Type

2)Area Name

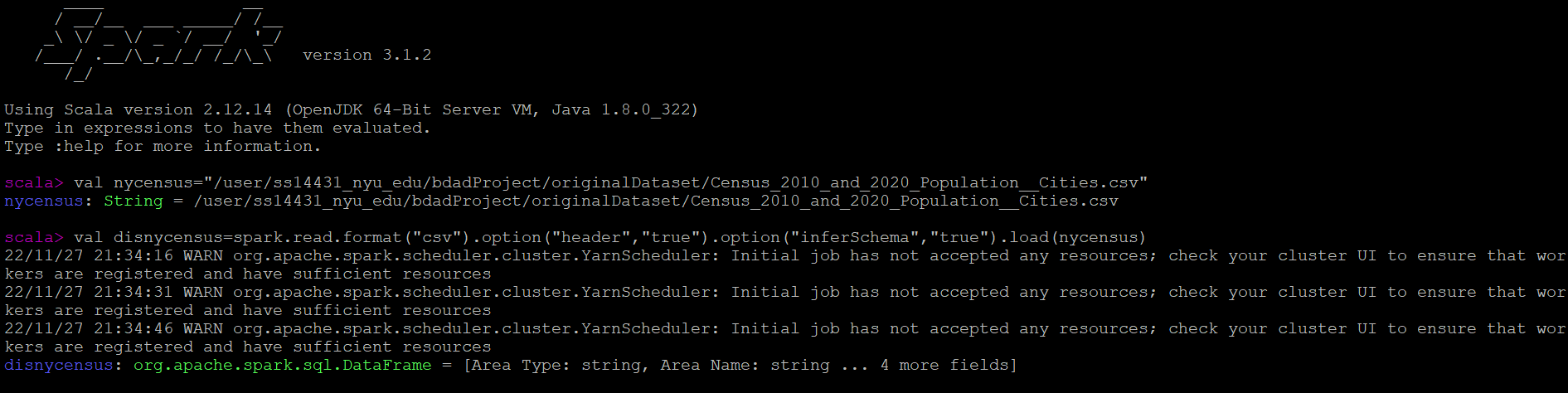
3)2010 Census

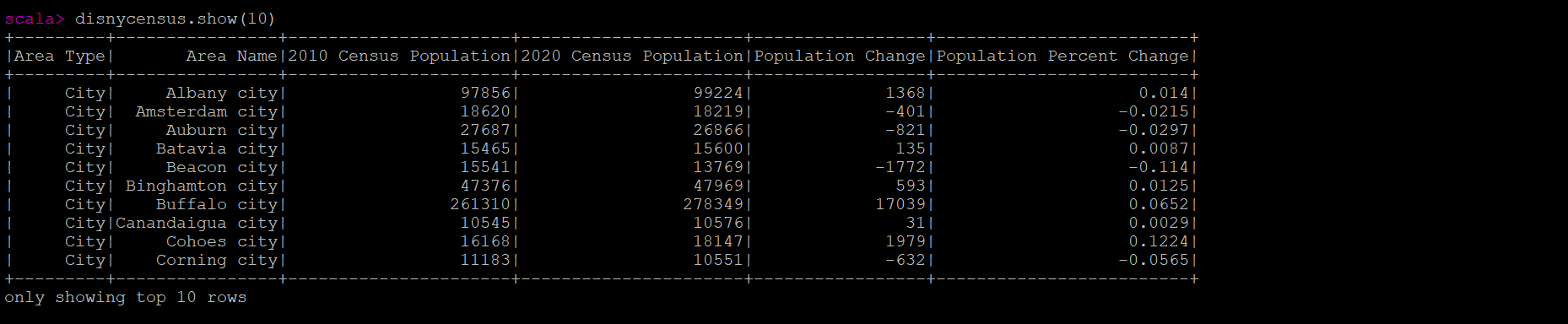
4)2020 Census

5)Population

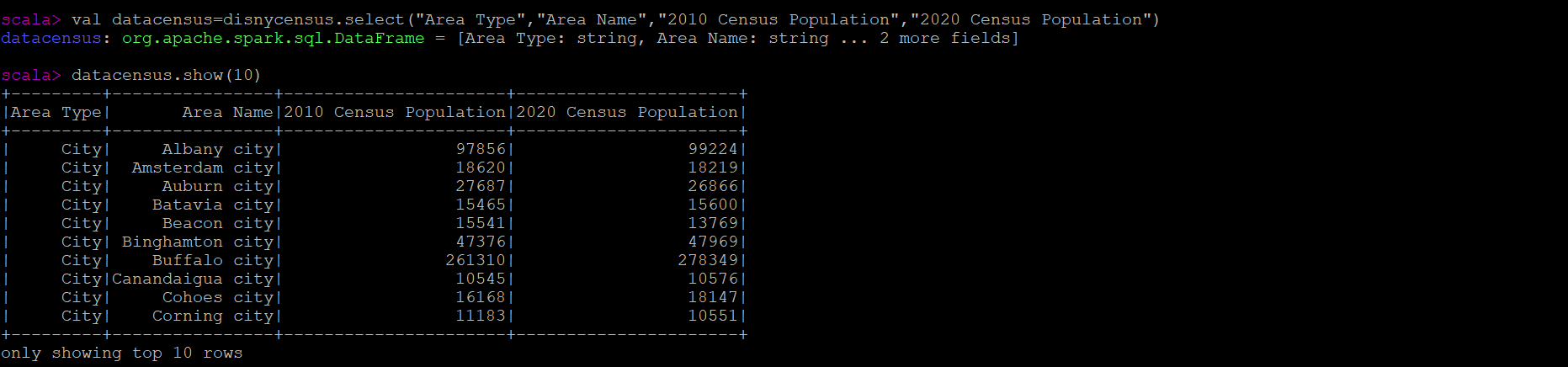
6)Percentage Change

The Census dataset is relatively clean as compared to the NYPD Complaints Dataset.Hence only unnecessary columns were dropped and the dataset was loaded to Hadoop for further use.





**Dropping Unnecessary Columns and then saving the file**



**Saved cleaned dataset in Hadoop for further use**:



**Overview of Scala Code used in the above Dataset :**

1)val nycensus="/user/ss14431\_nyu\_edu/bdadProject/originalDataset/Census\_2010\_and\_2020\_Population\_\_Cities.csv"

2)val disnycensus=spark.read.format("csv").option("header","true").option("inferSchema","true").load(nycensus)

3)disnycensus.show(10)

4)val datacensus=disnycensus.select("Area Type","Area Name","2010 Census Population","2020 Census Population")

5)datacensus.show(10)

6)Saving the file for further Use:

datacensus.write.format("parquet").save("/user/ss14431/bdadProject/cleaneddataset/Census.parquet")

**Note**:This is not the final processing for Census Data.More processing will be required when the demand arises.