CS432/532: Final Project Report

Project Title: Analysis on New York city Accidents

Team Member(s): 1. Samsher Gadkary [B00972197]
2. Sai Uday Nagula [B00979057]

I. Problem

Aim is to work on Motor Vehicle Collisions – Crashes dataset of New York City to analyze the collisions/crashes and discover the important information like accident prone areas, Time and date of occurrence, Contributing factors and types of vehicles involved, to take the safety measures.

The Problem Statements for which we performed analysis are as follows:

A. *We have made analysis on the data to compute the count of Accidents in different locations of New York city to take safety measures in those areas in different seasons and at different times in the day. This information can be used to identify the locations in which people must be more alert while driving.*

B. *We have done analysis on the several contributing factors to interpret the severity of each contributing factor based on the count of number of people killed or injured in an accident. This can be extremely useful while prioritizing the reasons behind accidents and focus on highly severe factors before we do focus on other factors.*

C. *We have done Analysis to figure out the percentage increase/decrease of accidents in each borough in New York city from year 2018 to 2022. This analysis can be used to interpret how well the authorities were able to control the number of accidents in New York city.*

Dataset Used for Analysis:
https://dev.socrata.com/foundry/data.cityofnewyork.us/h9gi-nx95

II. Software Design and Implementation

A. NoSQL-Database: MongoDB

We have used Mongo DB to store and run the analysis on the dataset. We have also used Mongo DB Compass which provides a very good interface to interact with Data Base, for visualization and to run queries. Mongo DB compass provides simple procedure to run queries with predefined function names like $match, $addfields, $sort etc.

B. Programming Language used: Python

We chose python programming language in our project because we are familiar with the language and it also provides **pymongo** library which allows to easily interact with MongoDB Databases and perform various database related tasks. PyMongo provides an API that closely mirrors the MongoDB query language and supports all major features of the database.

C. IDE used: Visual Studio Code

D. Tools used for Visualization: Matplotlib library (python)

Software design:

- We have installed MongoDB community server in our local system along with MongoDB compass.
- We have uploaded our dataset into our database using MongoDB compass.
- Then we connected to MongoDB from our code using pymongo library and used pymongo inbuilt functions like match, sort, etc. to create complex queries to perform analysis.
- We then made Visualization in terms of Pie chart, Multiple bar graphs using Matplotlib library in python.

III. Project Outcome

Analysis 1: Correlation of Seasons and Accidents
This is the analysis of correlation between Accidents and at different times in a day in different seasons in each Borough in New York city.

```
C:\Users\nagul\Desktop\Spring 2023\DB\project_NoSql\code>py first1.py
Crashes happening in Winter
   Time        BROOKLYN    BRONX    MANHATTAN    QUEENS     STATEN ISLAND
04  -   12       30893     15173      20416      27371          4168
12  -   20       52779     24213      34360      44895          7773
20  -   04       20665     9417       16661      17253          2508

Crashes happening in Spring
   Time        BROOKLYN    BRONX    MANHATTAN    QUEENS     STATEN ISLAND
04  -   12       29944     14495      20433      25883          3677
12  -   20       53256     24358      36585      43571          7248
20  -   04       20447     9779       16473      16576          2363

Crashes happening in Summer
   Time        BROOKLYN    BRONX    MANHATTAN    QUEENS     STATEN ISLAND
04  -   12       29260     14133      21827      25523          3606
12  -   20       58696     26656      40711      48388          8071
20  -   04       24521     11840      18522      20225          2813

Crashes happening in Fall
   Time        BROOKLYN    BRONX    MANHATTAN    QUEENS     STATEN ISLAND
04  -   12       32935     15848      22958      29462          4235
12  -   20       56933     25221      40881      48122          8288
20  -   04       22710     10460      19209      19147          2595
```

- Brooklyn and Bronx are having more incidents of accidents in summer when compared to other seasons.
- Manhattan, Staten Island and Queens are having more incidents of accidents in Fall when compared to other seasons.

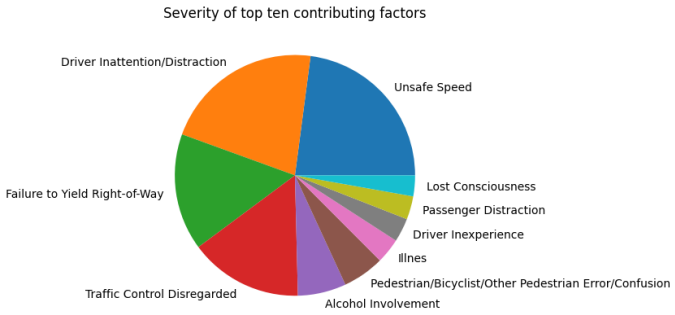
Analysis 2: Severity of top Ten Contributing Factors

This is the result of analysis of determining top ten most severe contributing factor of accidents sorted in the decreasing order of total number of people killed in accidents.

```
C:\Users\nagul\Desktop\Spring 2023\DB\project_NoSql\code>py second.py

People_Killed    People_Injured  Contributing Factor
     352         16732           Unsafe Speed
     331         122954          Driver Inattention/Distraction
     241         56630           Failure to Yield Right-of-Way
     234         22965           Traffic Control Disregarded
     100         10098           Alcohol Involvement
     86          6639            Pedestrian/Bicyclist/Other Pedestrian Error/Confusion
     52          1363            Illnes
     49          8850            Driver Inexperience
     48          4049            Passenger Distraction
     43          3486            Lost Consciousness
```

Pie Chart view of this analysis:



- Most Common reason behind more people being killed in accidents seems to be driving with Unsafe Speed or Drivers inattention/ Distractions.
- The next most Common reason are Failure to Yield Right-of-way or Traffic Control Disregarded.

Analysis 3: Percentage Change of number of Accidents

This is the analysis of change in number of accidents in each borough from the year 2018 to 2022.

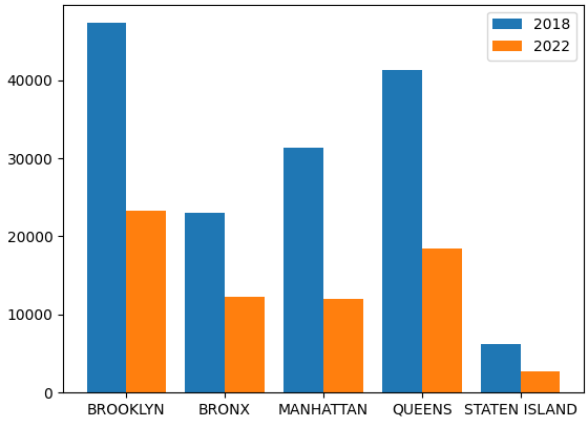
```
C:\Users\nagul\Desktop\Spring 2023\DB\project_NoSql\code>py third.py

Percentage of change in Accidents of each region from 2018 to 2022.

Accidents in BROOKLYN has changed from  47313  to  23337  by  -50.68 %
Accidents in BRONX has changed from  23060  to  12278  by  -46.76 %
Accidents in MANHATTAN has changed from  31412  to  11963  by  -61.92 %
Accidents in QUEENS has changed from  41278  to  18422  by  -55.37 %
Accidents in STATEN ISLAND has changed from  6171  to  2719  by  -55.94 %

Note: Negative sign in percentage say that there has been decrease in the Accidents
```

Multiple Bar graph:



- It is clear that Authorities in all the 5 Boroughs in New York city were able to successfully decrease the number of accidents each year from 2018 to 2022.
- Among all the Boroughs, Manhattan was able to decrease the numbers of accidents by most percentage.

SOURCE CODES

Code for Analysis 1:

```
import pymongo

myclient = pymongo.MongoClient('mongodb://localhost:27017/')
mydb = myclient['Binghamton']
db = mydb["Accidents"]

def analy(mn,hr):
  query = [
       {
    "$addFields": {
     "date": { "$toDate": "$CRASH DATE" }
    }
  },
  {
    "$project": {
       "bo": "$BOROUGH",
     "year": { "$year": "$date" },
     "month": { "$month": "$date"},
     "hour": { "$toInt": { "$arrayElemAt": [ { "$split": [ "$CRASH TIME", ":" ] }, 0 ] } }
    }
  },
  {"$match":
          {
             "month": {"$in": mn},
             "hour": { "$in": hr }

          }
    },
    {'$group':
          {
             "_id" : "$bo", "val": {"$sum":1}
          }
    },
    {
         '$sort' : {"val": -1}
    }
    ]

  query_ = db.aggregate(query)

  for w in query_:
    if(w.get('_id')=="BROOKLYN"): wd1 =(w.get('val'))
    if(w.get('_id')=="BRONX"): wd2 =(w.get('val'))
    if(w.get('_id')=="MANHATTAN"):wd3 =(w.get('val'))
    if(w.get('_id')=="QUEENS"): wd4 =(w.get('val'))
    if(w.get('_id')=="STATEN ISLAND"): wd5 =(w.get('val'))

  print('{:02d}'.format(hr[0])," -  ",'{:02d}'.format(hr[7]+1), "\t",wd1, "\t  ",wd2,"     ",wd3," \t",wd4," \t",wd5)

hr1 = [[4,5,6,7,8,9,10,11], [12,13,14,15,16,17,18,19], [20,21,22,23,0,1,2,3]]

mn1 = [12,1,2]
mn2 = [3,4,5]
mn3 = [6,7,8]
```

```
mn4 = [9,10,11]

print('Crashes happening in Winter')
print("   Time       BROOKLYN    BRONX    MANHATTAN    QUEENS    STATEN ISLAND")
for i in hr1:
  analy(mn1,i)

print('\nCrashes happening in Spring')
print("   Time       BROOKLYN    BRONX    MANHATTAN    QUEENS    STATEN ISLAND")
for i in hr1:
  analy(mn2,i)

print('\nCrashes happening in Summer')
print("   Time       BROOKLYN    BRONX    MANHATTAN    QUEENS    STATEN ISLAND")
for i in hr1:
  analy(mn3,i)

print('\nCrashes happening in Fall')
print("   Time       BROOKLYN    BRONX    MANHATTAN    QUEENS    STATEN ISLAND")
for i in hr1:
  analy(mn4,i)
```

Code for Analysis 2:

```
import pymongo

myclient = pymongo.MongoClient('mongodb://localhost:27017/')
mydb = myclient['Binghamton']
db = mydb["Accidents"]

contributing_factors = [
  {
    "$project": {
       "people_killed" : { "$add" : ["$NUMBER OF PEDESTRIANS KILLED", "$NUMBER OF CYCLIST KILLED",
"$NUMBER OF MOTORIST KILLED"]},
       "people_injured" : { "$add" : ["$NUMBER OF PEDESTRIANS INJURED", "$NUMBER OF CYCLIST INJURED",
"$NUMBER OF MOTORIST INJURED"]},
       "co" : "$CONTRIBUTING FACTOR VEHICLE 1"
    }
  },
    {'$group':
          {
             "_id" : "$co" , "val": {"$sum":"$people_killed"} , "val1": {"$sum":"$people_injured"}
          }
    },
    {
          '$sort' : {"val": -1}
    },
    {
       "$skip" : 1
    },
```

```
    {
       "$limit" : 10
    }
]

people= []
reason = []
contributing_factors_ = db.aggregate(contributing_factors)
print("\nPeople_Killed    People_Injured  Contributing Factor")
for w in contributing_factors_:
    people.append(w.get("val"))
    reason.append(w.get("_id"))
    print('    ',w.get("val"),'     \t',w.get("val1"),'     \t', w.get("_id").strip())

# CODE FOR VISULAIZATION USING MATPLOTLIB
import matplotlib.pyplot as plt

# CODE FOR PIE CHART
fig,ax = plt.subplots()
ax.set_title("Severity of top ten contributing factors")
ax.pie(people,labels=reason)
      plt.show()
```

Code for Analysis 3:

```
import pymongo

myclient = pymongo.MongoClient('mongodb://localhost:27017/')
mydb = myclient['Binghamton']
db = mydb["Accidents"]

year_2018 = [
    {
    "$addFields": {
      "date": { "$toDate": "$CRASH DATE" }
    }
  },
  {
    "$project": {
       "bo": "$BOROUGH",
      "year": { "$year": "$date" },
    }
  },
  {"$match":
          {
             "year": {"$in": [2018]},
          }
    },
    {'$group':
          {
             "_id" : "$bo", "val": {"$sum":1}
          }
    }
```

```
]

year_2022 = [
   {
   "$addFields": {
     "date": { "$toDate": "$CRASH DATE" }
   }
  },
  {
   "$project": {
      "bo": "$BOROUGH",
     "year": { "$year": "$date" }
   }
  },
  {"$match":
         {
            "year": {"$in": [2022]}
         }
   },
   {'$group':
         {
            "_id" : "$bo", "val": {"$sum":1}
         }
   }
]

year_2018_result = db.aggregate(year_2018)
year_2022_result = db.aggregate(year_2022)

for w in year_2018_result:
   if(w.get('_id')=="BROOKLYN"): brooklyn_2018 =(w.get('val'))
   if(w.get('_id')=="BRONX"): bronx_2018 =(w.get('val'))
   if(w.get('_id')=="MANHATTAN"):manhattan_2018 =(w.get('val'))
   if(w.get('_id')=="QUEENS"): queens_2018 =(w.get('val'))
   if(w.get('_id')=="STATEN ISLAND"): staten_island_2018 =(w.get('val'))

for w in year_2022_result:
   if(w.get('_id')=="BROOKLYN"): brooklyn_2022 =(w.get('val'))
   if(w.get('_id')=="BRONX"): bronx_2022 =(w.get('val'))
   if(w.get('_id')=="MANHATTAN"):manhattan_2022 =(w.get('val'))
   if(w.get('_id')=="QUEENS"): queens_2022 =(w.get('val'))
   if(w.get('_id')=="STATEN ISLAND"): staten_island_2022 =(w.get('val'))

print(("\nPercentage of change in Accidents of each region from 2018 to 2022.\n"))

print("Accidents in BROOKLYN has changed from ",brooklyn_2018," to ",brooklyn_2022," by ", round(((brooklyn_2022 - brooklyn_2018)/brooklyn_2018)*100,2),"%" )
print("Accidents in BRONX has changed from ",bronx_2018," to ",bronx_2022," by ", round(((bronx_2022 - bronx_2018)/bronx_2018)*100,2),"%" )
print("Accidents in MANHATTAN has changed from ",manhattan_2018," to ",manhattan_2022," by ", round(((manhattan_2022 - manhattan_2018)/manhattan_2018)*100,2),"%" )
print("Accidents in QUEENS has changed from ",queens_2018," to ",queens_2022," by ", round(((queens_2022 - queens_2018)/queens_2018)*100,2),"%" )
print("Accidents in STATEN ISLAND has changed from ",staten_island_2018," to ",staten_island_2022," by ", round(((staten_island_2022 - staten_island_2018)/staten_island_2018)*100,2),"%" )
print("\nNote: Negative sign in percentage say that there has been decrease in the Accidents")

# CODE FOR VISULAIZATION USING MATPLOTLIB
import matplotlib.pyplot as plt
```

```
import numpy as np
# CODE FOR MULTIPLE BAR GRAPH
la = ['BROOKLYN','BRONX','MANHATTAN','QUEENS',"STATEN ISLAND"]
X_axis = np.arange(len(la))
fig,ax = plt.subplots()
ax.bar(X_axis - 0.2,[brooklyn_2018,bronx_2018,manhattan_2018,queens_2018,staten_island_2018], 0.4,label='2018')
ax.bar(X_axis + 0.2,[brooklyn_2022,bronx_2022,manhattan_2022,queens_2022,staten_island_2022], 0.4,label='2022')
plt.xticks(X_axis, la)
ax.legend()
    plt.show()
```

REFERENCES

[1] https://www.mongodb.com/docs/manual/tutorial/query-documents/

[2] https://matplotlib.org/stable/plot_types/index.html.

[3] https://pymongo.readthedocs.io/en/stable/tutorial.html

[4] https://youtu.be/6_NSkDRXPZk