Problem Statement

Objectives of the Assignment

- Primarily, this assignment is meant as a deep dive into the usage of Spark. As you saw while working with Spark, its syntax behaves differently from a regular Python syntax. One of the major objectives of this assignment is to gain familiarity with how analysis works in PySpark.
- Learning the basic idea behind using functions in PySpark can help in using other libraries like SparkR. If you are in a company where R is a primary language, you can easily pick up SparkR syntax and use Spark's processing power.
- Learning some new Spark commands and applying them to solve a problem.

Problem Statement

Big data analytics allows you to analyse data at scale. It has applications in almost every industry in the world. Let's consider an unconventional application that you wouldn't ordinarily encounter.

New York City is a thriving metropolis. Just like most other metros its size, one of the biggest problems its citizens face is parking. The classic combination of a huge number of cars and cramped geography leads to a huge number of parking tickets.

In an attempt to scientifically analyse this phenomenon, the NYC Police Department has **collected data for parking tickets**. Of these, the data files for multiple years are publicly available on Kaggle. We will try and perform some **exploratory analysis** on a part of this data. Spark will allow us to analyse the full files at high speeds as opposed to taking a series of random samples that will approximate the population. For the scope of this analysis, we will analyse the parking tickets over the year **2017**.

Note: Although the broad goal of any analysis of this type is to have better parking and fewer tickets, we are **not looking for recommendations on how to reduce the number of parking tickets**—there are no specific points reserved for this.

The purpose of this assignment is to conduct an exploratory data analysis that will help you understand the data. Since the size of the dataset is large, your queries will take some time to run, and you will need to identify the correct queries quicker. The questions given below will guide your analysis.

The dataset structure is available *on this page* along with the data.

General Guidelines:

- If you make any specific assumptions related to these questions, be sure to state them.
- Include all the necessary commands to prevent errors.
- The queries may take time to get executed. Please have some patience. If you are getting errors with correct queries, restart the PySpark session and try again, as the session may have expired.
- Keep a copy of the commands on your local drive so that you do not lose any work in case of session expiry.
- If you want to run SQL commands, create an SQL view first. Also, if you make any changes in the table (like substitution or dropping null values), please ensure that you update the SQL view related to that table for further analysis.
- **Remember to stop the EC2 instance whenever you are done for the day.**

Questions to Be Answered in the Analysis

The following analysis should be performed on PySpark running on your EC2 instance, using the Jupyter Notebook. Remember that you need to summarise the analysis with your insights along with the code.

Examine the data

- Find the total number of tickets for the year.
- Find out the number of unique states from where the cars that got parking tickets came. *(**Hint**: Use the column 'Registration State'.)*

  There is a numeric entry '99' in the column, which should be corrected. Replace it with the state having the maximum entries. Provide the number of unique states again.

  *(**Hint**: you can edit entries by using the when-otherwise statement)*
- Display the top 20 states with the most number of tickets along with their ticket count.

Aggregation tasks

- How often does each violation code occur? Display the frequency of the top five violation codes.
- How often does each 'vehicle body type' get a parking ticket? How about the 'vehicle make'? Find the top 5 for both.
- Let's try and find some seasonality in this data:
  - First, divide the year into 4 seasons, and find the frequencies of tickets for each season. (***Hints**: Use Issue Date to segregate into seasons. You may use a UDF or when-otherwise statement to do so. You have to cast the date format before you can get the month of IssueDate.)*
  - Then, find the three most common violations for each of these seasons.

- The fines collected from all the instances of parking violation constitute a source of revenue for the NYC Police Department. Let's take an example of estimating this for the three most commonly occurring codes:
    - Find the total occurrences of the three most common violation codes.
    - Then, visit the website:

      http://www1.nyc.gov/site/finance/vehicles/services-violation-codes.page

      It lists the fines associated with different violation codes. They're divided into two categories: one for the highest-density locations in the city and the other for the rest of the city. For the sake of simplicity, take the average of the two.
    - Using this information, find the total amount collected for each of the three violation codes with the maximum tickets. State the code that has the highest total collection (only based on the top 3 tickets). *(**Hint:** It may be a wise idea to store the fines in a separate column, based on the violation code)*
    - Find the top 3 states that have the highest ticket revenue based on the top 3 violation codes alone. *(**Hint**: Use the column 'Registration State'.)*
    - What can you intuitively infer from these findings?