Problem Statement

Objectives:

Primarily, this case study 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 case study is to gain familiarity with how analysis works in PySpark as opposed to base Python.

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.

The process of running a model-building command boils down to a few lines of code. While drawing inferences from data, the most time-consuming step is preparing the data up to the point of model building. So, this case study will focus more on exploratory analysis.

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 case study is to conduct an exploratory data analysis that will help you understand the data. The questions given below will guide your analysis.

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.

- 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 Spark whenever you finish working on to the cluster. Use spark.stop().

Accessing the Dataset

The data for this case study has been placed in HDFS at the following path:

'/common_folder/nyc_parking/Parking_Violations_Issued_-_Fiscal_Year_2017.csv'

Questions to Be Answered in the Analysis

The following analysis should be performed on PySpark mounted on your CoreStack cluster, using the PySpark library. Remember that you need to summarise the analysis with your insights along with the code.

Examine the data

1. Find the total number of tickets for the year.
2. 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.

Aggregation tasks

1. How often does each violation code occur? Display the frequency of the top five violation codes.
2. How often does each 'vehicle body type' get a parking ticket? How about the 'vehicle make'? (*Hint*: *Find the top 5 for both.)*
3. A precinct is a police station that has a certain zone of the city under its command. Find the (5 highest) frequencies of tickets for each of the following:
   - 'Violation Precinct' (This is the precinct of the zone where the violation occurred). Using this, can you draw any insights for parking violations in any specific areas of the city?
   - 'Issuer Precinct' (This is the precinct that issued the ticket.)
     Here, you would have noticed that the dataframe has the'Violating Precinct' or 'Issuing Precinct' as '0'. These are erroneous entries. Hence, you need to provide the records for five correct precincts. (Hint: Print the top six entries after sorting.)

4. Find the violation code frequencies for three precincts that have issued the most number of tickets. Do these precinct zones have an exceptionally high frequency of certain violation codes? Are these codes common across precincts? *(Hint: In the SQL view, use the 'where' attribute to filter among three precincts.)*

5. Find out the properties of parking violations across different times of the day:
   - Find a way to deal with missing values, if any. *(Hint: Check for the null values using 'isNull' under the SQL. Also, to remove the null values, check the 'dropna' command in the API documentation.)*
   - The Violation Time field is specified in a strange format. Find a way to make this a time attribute that you can use to divide into groups.
   - Divide 24 hours into six equal discrete bins of time. Choose the intervals as you see fit. For each of these groups, find the three most commonly occurring violations.
     (Hint: Use the CASE-WHEN in SQL view to segregate into bins. To find the most commonly occurring violations, you can use an approach similar to the one mentioned in the hint for question 4.)
   - Now, try another direction. For the three most commonly occurring violation codes, find the most common time of the day (in terms of the bins from the previous part).

6. Let's try and find some seasonality in this data:

   - First, divide the year into a certain number of seasons, and find the frequencies of tickets for each season. *(Hint: Use Issue Date to segregate into seasons.)*
   - Then, find the three most common violations for each of these seasons. *(Hint: You can use an approach similar to the one mentioned in the hint for question 4.)*

7. 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 the three violation codes with the maximum tickets. State the code that has the highest total collection.
- What can you intuitively infer from these findings?