Monash University

FIT5202 - Data processing for Big Data

Assignment 1: Analyzing Road Crash Data

Due Date: **Sunday Sept 6, 2020, 11:55 PM (Local Campus Time)**
Worth: 10% of the final marks

Background

The Department of Planning, Transport and Infrastructure (DPTI), South Australia collects data from various road crashes for further analysis in an endeavor to improve road safety. Over time, the data increases in size; the increase in the number of vehicles also contributes to huge amounts of data. As we look across multiple states, we can imagine a rather large set of data. Here, we want to employ various operations on the dataset using Spark to answer different queries.

Required Datasets (available in Moodle):

- Two datasets:
  - Unit and Crash datasets from year 2015-2019
- A Metadata file is included which contains the information about the dataset.
- These files are available in Moodle under Assessment 1.

Information on Dataset

The data used here is the Road Crash Data from 2015-2019 for South Australia prepared by the Department of Planning, Transport and Infrastructure (DPTI) . The data is available on the website https://data.sa.gov.au.
The datasets contain various details about the crash events including the vehicle and the people involved in the crash. In this assignment, only two datasets i.e. Crash and Units are considered. For more detailed information on the dataset, **please refer to the Metadata file included in the assignment dataset or from the given website**.

Note: In the dataset, the exact day of the crash is not released by the data provider, being considered as sensitive information. When displaying dates, please use the format (**Year-Month-Dayofweek**) E.g. (2017-January-Sunday).

Assignment Information

This assignment consists of two parts:

- **Part A:** You are required to implement various solutions based on RDDs and DataFrames in PySpark for the given queries related to crash data analysis.
- **Part B:** You are required to create a video presentation discussing your observations for the questions in **Part A** (i.e. **1.2**, **2.3 & 2.4**).

Getting Started

- Download the datasets from moodle.
- Create an ***Assignment-1.ipynb*** file in jupyter notebook to write your solution.
- You will be using Python 3+ and PySpark 3.0 for this assignment.

Part A : Working with RDDs and DataFrames (80%)

---

1. Working with RDD (25%)

In this section, you will need to create RDDs from the given datasets, perform partitioning in these RDDs and use various RDD operations to answer the queries for crash analysis.

1.1 Data Preparation and Loading (5%)

1. Write the code to create a SparkContext object using SparkSession, which tells Spark how to access a cluster. To create a SparkSession you first need to build a SparkConf object that contains information about your application. Give an appropriate name for your application and run Spark **locally** with **as many working processors as logical cores on your machine**.
2. Import all the "Units" csv files from 2015-2019 into a single RDD.
3. Import all the "Crashes" csv files from 2015-2019 into a single RDD.
4. For each Units and Crashes RDDs, **remove the header rows** and **display the total count and first 10 records**. ***Hint :*** *You can use* ***csv.reader*** *to parse rows in RDDs.*

1.2 Data Partitioning in RDD (10%)

1. How many partitions do the above RDDs have? How is the data in these RDDs partitioned by default, when we do not explicitly specify any partitioning strategy?
2. In the "Units" csv dataset, there is a column called **Lic State** which shows the state where the vehicle is registered. Assume we want to keep all the data related to SA in one partition and the rest of the data in another partition.
    a. Create a Key Value Pair RDD with **Lic State** as the key and rest of the other columns as value.
    b. Write the code to implement this partitioning in RDD using appropriate partitioning functions.
    c. Write the code to print the number of records in each partition. What does it tell about the data skewness?

1.3 Query/Analysis (10%)

For the **Units** RDD, write relevant RDD operations to answer the following queries.

1. Find the average age of male and female drivers separately.
2. What is the oldest and the newest vehicle year involved in the accident? Display the Registration State, Year and Unit type of the vehicle.

2. Working with DataFrames (35%)

In this section, you will need to load the given datasets into PySpark DataFrames and use *DataFrame functions* to answer the queries.

2.1 Data Preparation and Loading (5%)

1. Load all units and crash data into two separate dataframes.
2. Display the schema of the final two dataframes.

2.2 Query/Analysis (15%)

Implement the following queries using dataframes. You need to be able to perform operations like filtering, sorting, joining and group by using the functions provided by the DataFrame API.

1. Find all the crash events in Adelaide where the total number of casualties in the event is more than 3.
2. Display 10 crash events with highest **casualties**.
3. Find the total number of **fatalities** for each crash type.
4. Find the total number of casualties for each suburb when the vehicle was driven by an unlicensed driver. You are required to display the name of the suburb and the total number of casualties.

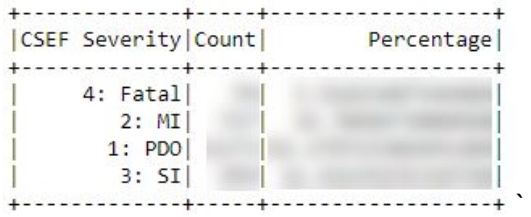
2.3 Severity Analysis (15%)

In this section, we want to analyze whether severity of accidents is higher when the driver is on drugs or alcohol compared to when the driver is normal. The severity of the crash is given by the column "CSEF Severity", the three levels of severity is given below (also included in the Metadata file). Similarly the columns "DUI Involved" and "Drugs Involved" tell whether the driver has been detected with blood alcohol and drugs respectively.

Crash Severity	Defines the road crash severity (classified by the highest injury severity sustained in the crash). Decoded values: "3: SI" = Serious Injury, "2: MI" = Minor Injury, "1: PDO" = Property Damage Only

Using the DataFrame for crash events, implement the following queries:

1. Find the total number of crash events for each severity level. Which severity level is the most common?
2. Compute the total number of crash events for each severity level and the percentage for the four different scenarios.
   A sample output for each of these scenarios is given below.



CSEF Severity	Count	Percentage
4: Fatal		
2: MI		
1: PDO		
3: SI		

   a. When the driver is tested positive on drugs.
   b. When the driver is tested positive for blood alcohol concentration.
   c. When the driver is tested positive for both drugs and blood alcohol
   d. When the driver is tested negative for both (no alcohol and no drugs).

   Compare the results in these 4 scenarios. Briefly explain the observation from this analysis.

2.4 RDDs vs DataFrame vs Spark SQL (20%)

Implement the following queries using RDDs, DataFrames and SparkSQL separately. Log the time taken for each query in each approach using the "%%time" built-in magic command in Jupyter Notebook and discuss the performance difference of these 3 approaches.

1. Find the Date[1] and Time of Crash, Number of Casualties in each unit and the Gender, Age, License Type of the unit driver for the suburb "Adelaide".
2. Find the total number of casualties for each suburb when the vehicle was driven by an unlicensed driver. You are required to display the name of the suburb and the total number of casualties.

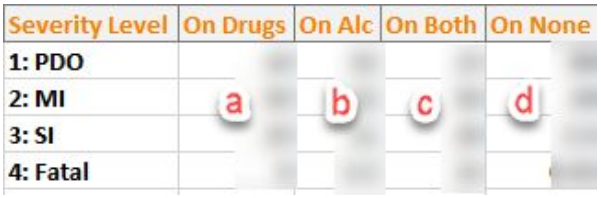
[1] When displaying dates, please use the format (**Year-Month-Dayofweek**) E.g. (2017-January-Sunday).

Part B : Pre-recorded Video Presentation (20%)

IMPORTANT: Pre-recorded video presentation is compulsory. No marks will be awarded if the assignment submission is missing the video.

For this part of the assignment, you are required to submit a pre-recorded video presentation as well. Your observations and analysis of the outputs to be included in the video presentation are as follows:

1. **RDD Partitioning** (**Task 1.2**): Discuss how you implemented the partitioning strategy. Also how is the data spread across the partitions after implementing the partitioning strategy? Do you see any skew in data, if so what other approaches can you use to manage the skew?
2. **Crash Severity Analysis (Task 2.3) :** Based on the results from 2.3.2, combine the results from the four different scenarios to a single table (as shown below) and visualize it with a bar graph. ***Hint***: *You can use matplotlib in python or you can use Excel to visualize the data.*



3. **RDDs vs DataFrame vs Spark SQL(Task 2.4):** Based on the results from 2.4 (1 & 2), present your findings. Explain which approach works faster and the possible reasons for the same.

Further Information for Video Presentation

- The Instructions for Recording Video Presentation can be found here.
- Create 3-5 powerpoint slides (ideally 1 slide for each topic).
- Record the video using zoom screen sharing and record feature.
    - **IMPORTANT** : Please make sure you turn on your camera while recording the video presentation.
- Keep the video length from 4-6 mins.

Assignment Marking

The marking of this assignment is based on quality of work that you have submitted rather than just quantity. The marking starts from zero and goes up based on the tasks you have successfully completed and it's quality for example how well the code submitted follows *programming standards*, *code documentation*, *presentation of the assignment*, *readability of the code*, *organisation of code and so on*. Please find the PEP 8 -- Style Guide for Python Code here for your reference.
Your video presentation will be assessed on the basis of the overall quality of your presentation which includes the content and quality of the slides, the observation and explanation presented and the quality of the delivery.

Submission

You should submit your final version of the assignment solution online via Moodle; You must submit the following:

- A zip file of your Assignment 1 folder, named based on your authcate name (e.g. psan002). This should contain
    - **Assignment-1.ipynb**
    - **Assignment-1 Video.mp4** .
      This should be a ZIP file and *not any other kind of compressed folder (e.g. .rar, .7zip, .tar)*. Please do not include the data files in the ZIP file.
- The assignment submission should be uploaded and finalised by Sunday September 6th, 11:55 PM (Local Campus Time).
- Your assignment will be assessed based on the contents of the Assignment 1 folder you have submitted via Moodle. When marking your assignments, we will use the same ubuntu setup as provided to you in Week 01.

Other Information

Where to get help

You can ask questions about the assignment on the Assignments section in the Ed Forum accessible from the on the unit's Moodle Forum page. This is the preferred venue for assignment clarification-type questions. You should check this forum regularly, as the responses of the teaching staff are "official" and can constitute amendments or additions to the assignment specification. Also, you can visit the consultation sessions if the problem and the confusions are still not solved.

Plagiarism and collusion

Plagiarism and collusion are serious academic offences at Monash University. Students must not share their work with any other students. Students should consult the policy linked below for more information.

https://www.monash.edu/students/academic/policies/academic-integrity

See also the video linked on the Moodle page under the Assignment block.

Students involved in collusion or plagiarism will be subject to disciplinary penalties, which can include:

- The work not being assessed
- A zero grade for the unit
- Suspension from the University
- Exclusion from the University

Late submissions

Late Assignments or extensions will not be accepted unless you submit a special consideration form. ALL Special Consideration, including within the semester, is now to be submitted centrally. This means that students MUST submit an online Special Consideration form via Monash Connect. For more details please refer to the **Unit Information** section in Moodle.

There is a **5% penalty per day including weekends** for the late submission.