Monash University

FIT5202 - Data processing for Big Data

Assignment 1: Analysing pedestrian traffic

Due: Sunday, Jan 17, 2021, 11:55 PM (Local Campus Time)
Worth: 10% of the final marks

Background

The council of the City of Melbourne collects various data for understanding the movement of people in the city in an endeavour to improve the traffic and facilitate better crowd control. In particular, they are using sensors to collect pedestrian counts at different locations. Over time, the data increases in size. Here, we want to employ various operations on the dataset using Spark to answer different queries.

Required Datasets (available in Moodle):

- Two data files
  - Pedestrian_Counting_System_-_Monthly_counts_per_hour.csv
  - Pedestrian_Counting_System_-_Sensor_Locations.csv
- A Metadata file is included which contains the information about the dataset.
- These files are available in Moodle under Assessment 1 data folder

Information on Dataset

Two data files from the City of Melbourne are provided, which captures the hourly count of pedestrians recorded by the sensors and the corresponding sensor locations. The data is also available on the website https://data.melbourne.vic.gov.au/.

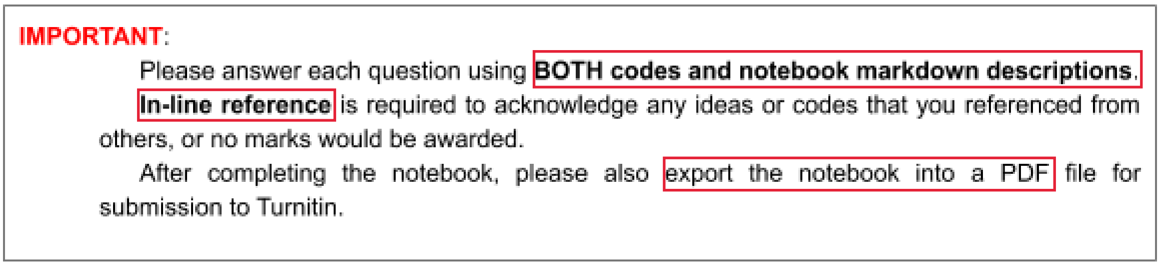
What you need to achieve

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 foot traffic data analysis.
- **Part B:** You are required to create a video presentation discussing your solutions and observations for some questions in **Part A** (i.e. **1.2.1**, **2.2.1 & 3.1**).

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.



IMPORTANT:

Please answer each question using **BOTH codes and notebook markdown descriptions**.

In-line reference is required to acknowledge any ideas or codes that you referenced from others, or no marks would be awarded.

After completing the notebook, please also export the notebook into a PDF file for submission to Turnitin.

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

1. Working with RDD (35%)

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 foot traffic analysis. **Please DO NOT read those files into the dataframe and convert it back to RDD**, otherwise, NO marks would be awarded in 1.1.2, 1.1.3, 1.2.

1.1 Data Loading (17%)

1. Write the code to get a SparkContext object from SparkSession. For creating the SparkSession, you need to use a SparkConf object to configure the Spark app with a proper application name, to use Melbourne as the session timezone, and to run locally with as many working processors as local cores on your machine[1]

2. Read pedestrian count CSV file into a single RDD; and read the sensor location CSV file into another RDD
   a. For each RDD, remove the header rows and parse each comma-delimited line into a Row object with each column following the data type from the metadata file, with the exception of the “location” column.
      i. The “location” column should be transformed into tuples of numeric data in the format of (xxx, xxx).
      ii. The “Date_Time” column should be converted to datetime format (no need to specify timezone).
      iii. The “installation_date” column should be converted to date format (no need to specify timezone)

3. For each RDD, display the number of columns, the total number of records, and display the first 2 records

1.2 Analysis (18%)

Write relevant RDD operations to answer the following queries.

1. In the pedestrian count dataset, there is a column called “Hourly_Counts”. Assuming we want to perform range partitioning based on this column with the bin size of 1000, e.g. ranges being 0~999, 1000~1999, 2000~2999, etc. Write the code to implement

[1] More information about Spark configuration can be found in https://spark.apache.org/docs/latest/configuration.html

this partitioning in RDD using appropriate partitioning functions, and then display the number of partitions and the number of records in each partition

2. What is the trend from 2009 till 2020 in terms of the yearly total pedestrian counts and the average daily pedestrian counts? Display the year, the total pedestrian count and the average daily pedestrian count for each year, following the ascending order of year

3. Write the RDD operations to create another RDD, which combines the sensor description from the sensor location data, and calculates the average hourly pedestrian count for each sensor, and sort the average count from highest to lowest. Display the sensor ID, sensor description and average hourly count for the top-5 sensors with the highest average pedestrian counts

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. Excessive usage of Spark SQL is discouraged. **Please DO NOT read those files into the RDD and convert it to dataframe**, otherwise, NO marks would be awarded in 2.1 & 2.2.

2.1 Data Loading (15%)

1. Define the data schema[2] for both pedestrian count CSV files and the sensor location file, following the data types suggested in the metadata file, with the exception of the "Date_Time", "installation_date", and "location" columns
    a. Use StringType for "Date_Time", "installation_date", "location" columns and transform them later.

2. Using predefined schema, load the pedestrian count CSV file into a dataframe, and load the sensor location CSV file into another dataframe

3. For the "Date_Time" and "installation_date", "location" columns, transform them into the proper format specified below
    a. For the "Date_Time" column, transform it into date-time format.
        i. For the records with ID 2853222 and 2853223 in pedestrian count data, display the two records' Date_Time before and after the transformation; describe what issues you have found and suggest what approaches could potentially remediate this issue.
    b. For the "installation_date" column, convert it into date format.
    c. For the "location" column, transform it into a column of numeric arrays.
    d. Print the schema of both dataframe after transformation

[2] Sample usage of schema for reading CSV file can be found in https://docs.databricks.com/data/data-sources/read-csv.html

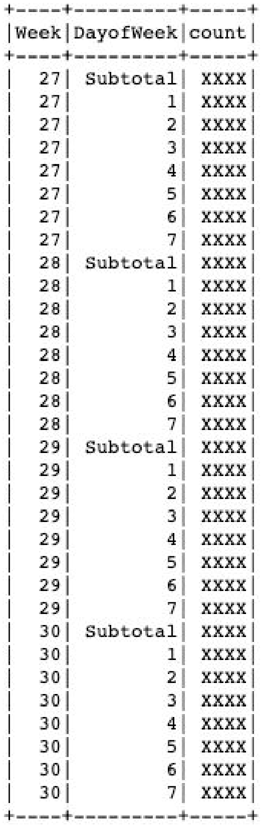
2.2 Analysis (20%)

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

1. Similar to 1.3.2, get the trend from 2009 to 2020 in terms of the yearly total pedestrian counts and the average daily pedestrian counts. Display the year, the total pedestrian count and the average daily pedestrian count for each year, following the ascending order of year

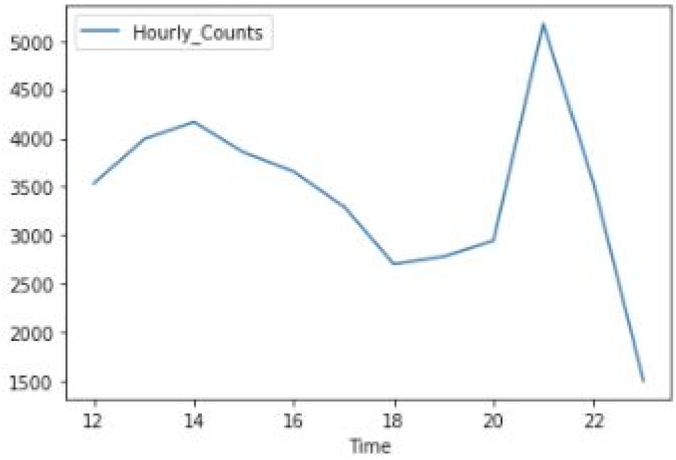
2. Similar to 1.3.3, display the sensor ID, sensor description and average hourly count for the top-5 sensors with the highest average hourly pedestrian counts. The dataframe created should compute the average hourly pedestrian count for each sensor and sort the average count from highest to lowest, while combining the sensor description from the sensor location data

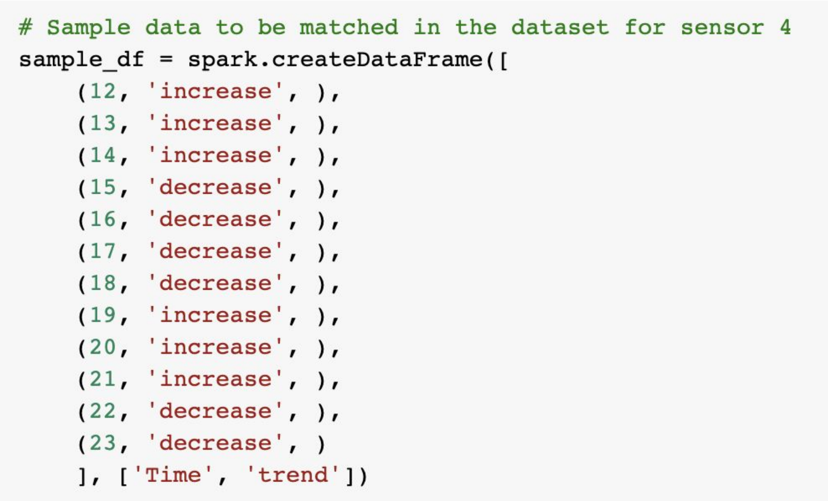
3. For the period starting from 2019-07-01 to 2019-07-28 (inclusive), get the breakdown of total daily pedestrian counts with the subtotal of the weekly counts. Your result should follow the format below, where “Week” column represents the week of year, “DayofWeek” column shows the subtotal or the day of week using the convention of Monday being the first day of week, and the “count” column shows the total pedestrian count of the day or the week. *Hint: you can use roll-up/cube operations*



```
+----+---------+-----+
|Week|DayofWeek|count|
+----+---------+-----+
|  27| Subtotal| XXXX|
|  27|        1| XXXX|
|  27|        2| XXXX|
|  27|        3| XXXX|
|  27|        4| XXXX|
|  27|        5| XXXX|
|  27|        6| XXXX|
|  27|        7| XXXX|
|  28| Subtotal| XXXX|
|  28|        1| XXXX|
|  28|        2| XXXX|
|  28|        3| XXXX|
|  28|        4| XXXX|
|  28|        5| XXXX|
|  28|        6| XXXX|
|  28|        7| XXXX|
|  29| Subtotal| XXXX|
|  29|        1| XXXX|
|  29|        2| XXXX|
|  29|        3| XXXX|
|  29|        4| XXXX|
|  29|        5| XXXX|
|  29|        6| XXXX|
|  29|        7| XXXX|
|  30| Subtotal| XXXX|
|  30|        1| XXXX|
|  30|        2| XXXX|
|  30|        3| XXXX|
|  30|        4| XXXX|
|  30|        5| XXXX|
|  30|        6| XXXX|
|  30|        7| XXXX|
+----+---------+-----+
```

4. Find all days when the sensor 4 exhibits the trend as below, in which the pedestrian count increases between 12:00 and 14:00, then decreases between 14:00 and 19:00, increases again between 19:00 and 21:00, and finally decreases until 23:00. For all the days you find with the patterns above, display the “Year”, “Month” and “Mdate”. Note the pattern only needs to follow the trend with sample data below, without the need to be in the same number range





```
# Sample data to be matched in the dataset for sensor 4
sample_df = spark.createDataFrame([
    (12, 'increase', ),
    (13, 'increase', ),
    (14, 'increase', ),
    (15, 'decrease', ),
    (16, 'decrease', ),
    (17, 'decrease', ),
    (18, 'decrease', ),
    (19, 'increase', ),
    (20, 'increase', ),
    (21, 'increase', ),
    (22, 'decrease', ),
    (23, 'decrease', )
    ], ['Time', 'trend'])
```

3 Performance comparison (10%)

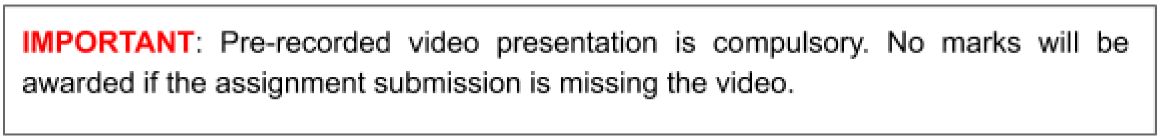
3.1 Performance comparison (5%)

1. Take screenshots from Spark UI for the job relating to the RDD operation in task 1.2.3 and for the job relating to the Dataframe operation in task 2.2.2. Your screenshots should show the time spent on each job, overall stage and task information for the job
2. Which one is faster? Explain the reasons behind the performance difference between the RDD job and Dataframe job

3.2 Performance improvement (5%)

1. For the task 2.2.4, take screenshots from Spark UI - SQL tab to show the query stages/lineage, as well as the details of logical plans and physical plan
2. What are the potential approaches that can improve this query performance? Discuss and give some examples to illustrate your ideas

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 1.2.1**): Discuss how you implemented the partitioning strategy. What are the potential problems using this partitioning strategy? Despite the skewness, can you think of a scenario where the partitioning on "Hourly_Counts" could be useful?
2. **Yearly Trend (Task 2.2.1):** Based on the results from 2.2.1, visualize it using matplotlib in python and discuss what you see from the plot (e.g. the trend, the abnormalities, and possible reasons behind them considering the sensor location data).
3. **Performance comparison (Task 3.1):** Based on the results from 3.1**,** present your findings and explain which approach works faster and the possible reasons.

Further Information for Video Presentation

- The Instructions for Recording Video Presentation can be found here. Note that the video has a different example, your slides need to be based on your use case.
- 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 the 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, reusability 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 PDF file (created from the notebook) to be submitted through Turnitin submission link
  - Use the browser's print function to save the notebook as PDF.

- A zip file of your Assignment 1 folder, named based on your authcate name (e.g. psan002). 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. Your ZIP file should only contain
  - **Assignment-1.ipynb**
  - **Assignment-1 Video.mp4**
- The assignment submission should be uploaded and finalised by Sunday January 17th, 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 (VM) as provided to you.

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. It is not permitted to ask assignment questions on commercial websites such as StackOverflow or other forms of forums.

You should check the Ed 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

There is a **10% penalty per day including weekends** for the late submission.

Note: Assessment submitted more than 7 calendar days after the due date will receive a mark of zero (0) for that assessment task. Students may not receive feedback on any assessment that receives a mark of zero due to late-submission penalty.

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.