Assignment 1 Specification

(Individual Assignment)

FIT 5202 - Data processing for Big Data

Due: Friday September 6, 2019, 11:55 PM (Local Campus Time)
Worth: 20% of the final marks

Part A: Analysing Text Data

In this part, we will look into modern software development methodologies. The everyday job of a data scientist / data engineer / software engineer involves working in a project team and following software development methodologies to develop software products or do data analysis. To prepare for such a career, it would be useful to have knowledge on the common software development methodologies. In this assignment, we will analyse two books on software development methodologies. We will see the distribution of words, the most common words in both books and the average frequency.. This approach can also be scaled to find the most common words and distribution of all words in the Internet.

Required Dataset (available in Moodle):

- **Book1:** "Agile Processes in Software Engineering and Extreme Programming.txt"
- **Book2:** "Scrum Handbook.txt"

Step 01: Import **pyspark** and initialize Spark

pyspark is the Spark Python API that exposes the Spark programming model to Python. **SparkContext** is the main entry point for Spark functionality. In Spark, communication occurs between a driver and executors. The driver has Spark jobs that it needs to run and these jobs are split into tasks that are submitted to the executors for completion. The results from these tasks are delivered back to the driver. In order to use Spark and its API, we will need to use a **SparkContext**. When running Spark, you start a new Spark application by creating a SparkContext. When the SparkContext is created, it asks the master for some cores to use to do the processing. The master sets these cores aside; they won't be used for other applications.

*Write the code to create a **SparkContext** object, which tells Spark how to access a cluster.* To create a SparkContext 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.

Step 02: Create Resilient Distributed Datasets (RDDs)

A **SparkContext** can be used to create Resilient Distributed Datasets (RDDs) on a cluster. *Write the code in pyspark to read the required dataset and display the total number of lines in each dataset.*

Step 03: Cleaning/Manipulating text.

Words should be counted independent of their case. So, you will have to change all words to lowercase. Further, if there are any leading or trailing spaces on a line, it should be removed.

Write a function that performs the following tasks on the RDDs:

1. *Removes all characters which are not alphabets except space(s).*
2. *Changes all upper case letters to lowercase..*
3. *Removes all leading or trailing spaces.*

You can use the python module **re** for matching patterns. Finally, display the contents of the RDDs after applying the function.

Step 04: Transforming the Data/Counting the words

Apply a transformation that will split each element of the RDD by its spaces and then create a word pairs for e.g. ('agile', 1), ('handbook', 1). Then, count the frequency of each word and *display the top 20 most frequent words*.

Step 05: Removing Stop Words

In computing, stop words are words which are filtered out before or after processing of natural language data. In natural language processing, useless words (data), are referred to as stop words. We would not want these words taking up space in our database, or taking up valuable processing time. For this, we can remove them easily, by storing a list of words that you consider to be stop words. **NLTK(Natural Language Toolkit)** in python has a list of stopwords stored in 16 different languages.

Write the code to remove stop words from the RDDs. You should use **nltk** package to remove the stop words. *Also find how many unique words do we now have in each RDD?*

Step 06: Find the average occurrence of a word

To find the average occurrence of a word, you will have to find the total number of words and divide that by the number of unique words.

Write the code to find the average occurrences of the words.

Step 7: Exploratory data analysis

Analyze the distribution of the words using the standard python library - **matplotlib.** Please make sure you are aware of the different factors such as *visual effects*, *coordinate system, labels*, *data type and scale* and *informative interpretation* before data visualisation . Also consider other aspects of visualization like *clarity*, *accuracy* and *efficiency* as well.

- *Compare the distribution of words in Book1 and Book2 The data spans many orders of magnitude and the features of the distribution not quite evident in the linear space. Therefore, use log scale (base 10) to plot the graph. Explain your understanding based on the graphs.*
- *Compare the top 15 most common words in Book1 and Book2. Plot the graphs and explain your understanding of the graph.*

Complete all the steps mentioned above in Part A and save the file as **Assignment 1 - Part A.ipynb**

Part B: Analysing CSV Data

In this part, you will analyze crime data from South Australia. The dataset reflects reported incidents of crime (suburb-based crime statistics for crimes against the person and crimes against property.) that occurred in South Australia since 2010.

Required Dataset (available in Moodle):

- Crime_Statistics_SA_2010_present.csv

Step 01: Import **pyspark** and initialize Spark

You will use **SparkContext** from **pyspark**, which is the main entry point for Spark Core functionality. The **SparkSession** object provides methods used to create **DataFrames** from various input sources. A **DataFrame** is equivalent to a relational table in Spark SQL and can be created using various functions in **SparkSession**. Once created, it can be manipulated using the various domain-specific-language (DSL) functions defined in DataFrame, Column.

*Write the code to create a **SparkContext** object, which tells Spark how to access a cluster.* To create a SparkContext 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. Write the code to create a **SparkSession** object that can be used to create the data frame from the input data source (CSV file).

Step 02: Create Dataframe

Write the code to create a data frame and provide the data source as the CSV file. How many records are there in the data frame?

Step 03: Write to Database

We will use MongoDB as our data source. Therefore, as a data loading step, you are required to *read the CSV file using spark session and insert all the records into MongoDB*. Use the `overwrite` mode when you are inserting the data.

Step 04: Read from Database

Create a Spark DataFrame to hold data from the MongoDB collection specified in the `spark.mongodb.input.uri` option which your SparkSession option is using. Display the schema of the data frame. **You will use this new data frame to perform all the steps mentioned below.**

Step 05: Calculate the statistics of numeric and string columns

Calculate the statistics of "Offence Count" and "Reported Date". *Find the count, mean, standard deviation, minimum and maximum for these attributes*. Explain with reasoning whether the minimum and maximum reported date is correct.

Step 06: Change the data type of a column

The `Date` column is in string format. You need to change it to date format using the user-defined functions (udf).

Step 07: Preliminary data analysis

Write the code to answer the following analytical queries.

- *How many level 2 offences are there? Display the list of level 2 offences.*
- *What is the number of offences against the person?*
- *How many serious criminal tresspasses with more than 1 offence count?*
- *What percentage of crimes are offences against the property?*

Step 08: Exploratory data analysis

Next, write code to analyze the following analytical queries and visualise it using the standard python library - `matplotlib`. Please make sure you are aware of the different factors such as *visual effects*, *coordinate system*, *data type and scale* and *informative interpretation* before data visualisation as well as you follow *clarity, accuracy* and *efficiency*.

- *Find the number of crimes per year. Plot the graph and explain your understanding of the graph.*
- *Find the number of crimes per month. Plot the graph and explain your understanding of the graph.*

- *Where do most crimes take place? Find the top 20 suburbs (which would also display postcode for e.g. Caulfield-3162)?. Plot the graph and explain your understanding of the graph.*
- *Find the number of serious criminal trespasses by day and month. Plot a graph and explain your understanding of the graph.*

Complete all the steps mentioned above in Part B and save the file as **Assignment 1 - Part B.ipynb**

Assignment Marking

The rubric for the assignment is available in the Moodle for your reference. 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.

References

Crimes Statistics Data: https://data.sa.gov.au/data/dataset/crime-statistics
Book 1: https://www.springer.com/gp/book/9783319916019
Book 2:
https://www.researchgate.net/publication/301685699_Jeff_Sutherland's_Scrum_Handbook

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 your **Assignment 1 - Part A.ipynb** and **Assignment 1 - Part B.ipynb** solution file. This should be a ZIP file and *not any other kind of compressed folder (e.g. .rar, .7zip, .tar).*
- The assignment submission should be uploaded and finalised by Friday 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 Assignment Discussion Forum on the unit's Moodle page. This is the preferred venue for assignment clarification-type questions. You should check this forum (and the News 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

Submission must be made by the due date otherwise penalties will be enforced. You must negotiate any extensions formally with your campus unit lecturer via the in-semester special consideration process:

http://www.monash.edu.au/exams/special-consideration.html

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