**ICP1**

**BIG DATA ANALYTICS AND APPLICATIONS(CS5542)**

*Sravani Niharika Garapati(16328843)*

*Nikhila Chirumamilla(16323180)*

**QUESTION:**

Write a spark program to group the words in a given text file (i.e., icp.txt)

**What I have learned from this ICP:**

In this ICP, Professor asked us to write a spark program to group words in a file. For this program to be done, there is a need of installing Spark and Hadoop in google colab. We also set up JVM(Java Virtual Machine). Then we have used many spark functions with python by importing PYSPARK library. Some of the functions, I have learned and used in this program are as shown below.

**SparkContext**: This is an entry point for spark functionality. It depicts the connection to a spark cluster and also used to create RDDs on that cluster.

**SparkConf()**: It configures the Spark application and is used to set up various Spark parameters as key-value pairs .

**textFile()**: This method is used to read a text file from HDFS or any other Hadoop supported file system and it reads every line in file as an element into RDD(Resilient Distributed Datasets)

**take ()**: This method is used to display specified number columns in a dataset, generally acts like a print function in python

**translate():** It returns an expression by replacing all the specified characters to the another given set of characters.

**maketrans():** It constructs a transition tables which specifies the list of characters that need to be deleted or replaced from a string.

**flatMap():** It generates new array which is formed by applying given function to all the elements in the array and then flatten the result by one level.

**Lambda():** It evaluates an expression for a given argument.

**Map()**: It is an RDD transformation which returns new RDD by applying transformation function(Lambda) on every element of RDD.

**Filter():** It is used to filter the rows in the RDD based on given condition.

**reduceByKey():** It reduces the value for each key using associative reduce function PySpark RDD.

**sortByKey()**: It is used to sort the values of the key in either ascending or descending order.

**saveAsTextFile()**: It is used to store our result in a directory.

**ICP Description about the Task:**

In this task, we need to write a spark program to group words in a text file based on their first letter. This is done in order of below mentioned steps:

* Installed Spark, Hadoop and also PySpark libraries and set up JVM(Java Virtual Machine).
* Configured Spark application by importing SparkContext and SparkConf and created an RDD
* Imported the required file to Google Colab and read the contents of the file into RDD using textFile() function
* Removed punctuations and whitespaces from the file in a User Defined Function(UDF) by using translate() and maketrans(), flatMap() functions.
* Removed numericals from the data and also deleted duplicates in the data using isnumeric() and distinct() functions respectively.
* Made first letter of every word into capital letter and used reduceByKey() function to group words based on their first letter.
* Then, by using sortByKey() function, sorted the words in the reverse alphabetical order
* Lastly, saved the output in a text file using saveAsTextFile() function.

**Challenges faced in this ICP:**

The challenges which we faced while doing this ICP are listed below:

* It became very tough for us to understand in detail about SparkContext, SparkConf and SparkSession and differences between and also which is we should use at a particular problem.
* Removing Punctuations and Whitespaces from the given text became hard for us until we used translate() and makeTrans() functions.
* Faced trouble in grouping similar words based on first letter and later reduceByKey() function made it easier.

**Screenshots of Execution of Code:**

**Graphical user interface, text, application, email

Description automatically generated**

*Installing Spark and Hadoop and setting up JVM*

**Graphical user interface, text, application, email

Description automatically generated**

*Setting up locations for Java and Spark and installing PySpark library*

**Graphical user interface, text, application

Description automatically generated**

*Configuring Spark application using SparkContext and SparkConf(6)*

**Graphical user interface, text, application, email

Description automatically generated**

*Loading the file into RDD and Writing a function to remove punctuations*

**Graphical user interface, text, application, email

Description automatically generated**

*Removing numericals from the data*

**Graphical user interface, text, application

Description automatically generated**

*Removing Duplicates from the data*

**Graphical user interface, text, application

Description automatically generated**

*Making first letter of every word to capital*

**Graphical user interface, text, application, email

Description automatically generated**

*Matching similar words by first character*

**Graphical user interface, text, application

Description automatically generated**

*Sorting the list in reverse alphabetical order*

**Graphical user interface, text, application, email

Description automatically generated**

*The final output file*

**Graphical user interface, text, application, email

Description automatically generated**

**Output File:**

****

**Video Link:**

*https://www.youtube.com/watch?v=sCPn31iQIvA*