Big Data Tools and Techniques

Assignment 01

Group 13

Group 13

@00783884	Mr	Nithin Puthan Veettil Kongadan
@00792908	Mr	Muhammad Hassan Mukhtar
@00804155	Mr	Mayur Kumar (**Absent**)
@00804635	Mr	Sachin .
@00823251	Mr	Mohammad Reza Haghighatju

Uploading the Data

Extracted emails.csv to emails file

Uploaded emails.csv, stopwords file to emails folder in databricks

View that all files in place with **dbutils.fs.ls("/FileStore/tables/emails")**

Loading Data

Data is loaded to **Dataframe**

With auto infer Schema Allowed

And reading Multilines

Also “ as end of field but double “” **should be counted as one** “

quote="", escape="", respectively

Data Exploration

Lets see how many columns data have?

['file', 'message']

Now what is composition of each column is

After observation we can see that this message can be parsed into many columns

Like message_id, to, form etc

Parsing Emails

Before Parsing Emails let us defined a schema

*For the time being all the sub postentional columns are given data type **string***

Let us define a user defined function which can be used for data splitting into tabular form **parse_email(message):**

Continue …

Parsing Emails

After that a User Defined Function **UDF** is used which make user defined function work in Spark Environment with Spark DataFrames.

In last after parsing columns are rename for readability of code.

Continue …

Parsing Emails

Now we have 17 columns in our DataFrame as follows

['file', 'Message-ID', 'Date', 'From', 'To', 'Subject', 'Mime-Version', 'Content-Type',

'Content-Transfer-Encoding', 'X-From', 'X-To', 'X-cc', 'X-bcc', 'X-Folder', 'X-Origin',

'X-FileName', 'Body']

Saved **DataFrame** in CSV for future uses

Q. Group A | Easy | 1.

@00804635 Mr Sachin .

To calculates key statistics about email senders from the df_with_parsed_statistics DataFrame.

Q. Group A | Easy | 1.

@00804635 Mr Sachin .

To calculates key statistics about email senders from the df_with_parsed_statistics DataFrame.

1. Count the total number of emails (total_emails) by selecting the "From" column and applying the count() method.

Q. Group A | Easy | 1.

@00804635 Mr Sachin .

To calculates key statistics about email senders from the df_with_parsed_statistics DataFrame.

1. *Count the total number of emails (total_emails) by selecting the "From" column and applying the count() method.*
2. *Count the number of unique email senders (unique_senders) by selecting the "From" column and applying the distinct().count() method.*

Q. Group A | Easy | 1.

@00804635 Mr Sachin .

1. *Count the total number of emails (total_emails) by selecting the "From" column and applying the count() method.*
2. *Count the number of unique email senders (unique_senders) by selecting the "From" column and applying the distinct().count() method.*
3. *Calculate the mean number of emails sent per sender (mean_emails_per_sender) by dividing the total number of emails by the number of unique senders.*

Q. Group B | Medium | 1.

@00823251 Mr Mohammad Reza Haghighatju

For identifying the top 10 email senders from the df_with_parsed_message DataFrame.

Q. Group B | Medium | 1.

@00823251 Mr Mohammad Reza Haghighatju

For identifying the top 10 email senders from the df_with_parsed_message DataFrame.

1. *Filters data to include only valid email addresses in the "From" column.*

Q. Group B | Medium | 1.

@00823251 Mr Mohammad Reza Haghighatju

For identifying the top 10 email senders from the df_with_parsed_message DataFrame.

1. *Filters data to include only valid email addresses in the "From" column.*
2. *Groups filtered data by the "From" column and counts the number of emails sent by each sender.*

Q. Group B | Medium | 1.

@00823251 Mr Mohammad Reza Haghighatju

1. *Filters data to include only valid email addresses in the "From" column.*
2. *Groups filtered data by the "From" column and counts the number of emails sent by each sender.*
3. *Sorts the resulting DataFrame in descending order by email count and limits to the top 10 senders.*

Q. Group B | Medium | 3.

@00783884 Mr Nithin Puthan Veettil Kongadan

Here we have to extracts the domain from the "From" and "To" email addresses in the df_with_parsed_message DataFrame, and then counts the number of emails sent from and to the specific domain 'enron.com'.

Q. Group B | Medium | 3.

@00783884 Mr Nithin Puthan Veettil Kongadan

Here we have to extracts the domain from the "From" and "To" email addresses in the df_with_parsed_message DataFrame, and then counts the number of emails sent from and to the specific domain 'enron.com'.

1. *Extracts the domain from the "From" and "To" email addresses using the split function.*

Q. Group B | Medium | 3.

@00783884 Mr Nithin Puthan Veettil Kongadan

Here we have to extracts the domain from the "From" and "To" email addresses in the df_with_parsed_message DataFrame, and then counts the number of emails sent from and to the specific domain 'enron.com'.

1. *Extracts the domain from the "From" and "To" email addresses using the split function.*
2. *Filters the data to count the emails sent from and to the specified domain 'enron.com'.*

Q. Group B | Medium | 3.

@00783884 Mr Nithin Puthan Veettil Kongadan

1. *Extracts the domain from the "From" and "To" email addresses using the split function.*
2. *Filters the data to count the emails sent from and to the specified domain 'enron.com'.*
3. *Prints the number of emails sent from, to, and the total exchanged with the 'enron.com' domain.*

Q. Group C | Difficult | 1.

@00792908 Mr Muhammad Hassan Mukhtar

Now we analyzes the subject lines of emails to identifies the top 100 most frequently occurring words. It uses Spark SQL functions to:

Q. Group C | Difficult | 1.

@00792908 Mr Muhammad Hassan Mukhtar

Now we analyzes the subject lines of emails to identifies the top 100 most frequently occurring words. It uses Spark SQL functions to:

1. *Filter out rows with null subject lines*

Q. Group C | Difficult | 1.

@00792908 Mr Muhammad Hassan Mukhtar

Now we analyzes the subject lines of emails to identifies the top 100 most frequently occurring words. It uses Spark SQL functions to:

1. *Filter out rows with null subject lines*
2. *Convert subject lines to lowercase and remove non-alphabetic characters*

Q. Group C | Difficult | 1.

@00792908 Mr Muhammad Hassan Mukhtar

Now we analyzes the subject lines of emails to identifies the top 100 most frequently occurring words. It uses Spark SQL functions to:

1. *Filter out rows with null subject lines*
2. *Convert subject lines to lowercase and remove non-alphabetic characters*
3. *Split subject lines into individual words*

Q. Group C | Difficult | 1.

@00792908 Mr Muhammad Hassan Mukhtar

1. *Filter out rows with null subject lines*
2. *Convert subject lines to lowercase and remove non-alphabetic characters*
3. *Split subject lines into individual words*
4. *Filter out empty strings and stopwords*

Q. Group C | Difficult | 1.

@00792908 Mr Muhammad Hassan Mukhtar

1. *Filter out rows with null subject lines*
2. *Convert subject lines to lowercase and remove non-alphabetic characters*
3. *Split subject lines into individual words*
4. *Filter out empty strings and stopwords*
5. *Count the occurrences of each word*

Q. Group C | Difficult | 1.

@00792908 Mr Muhammad Hassan Mukhtar

1. *Filter out rows with null subject lines*
2. *Convert subject lines to lowercase and remove non-alphabetic characters*
3. *Split subject lines into individual words*
4. *Filter out empty strings and stopwords*
5. *Count the occurrences of each word*
6. *Sort the words by count in descending order and limit to the top 100*

Thanks You

Contacts

N.PuthanVeettilKongadan@edu.salford.ac.uk

M.H.Mukhtar254@edu.salford.ac.uk

M.R.Haghighatju@edu.salford.ac.uk

M.Kumar8@edu.salford.ac.uk

Sachin2@edu.salford.ac.uk