

“Lack of Student Engagement in Group Projects”

Analysis Project

Tyler Rollinson & Gabrael Smallwood

University of North Texas

IPAC 4240: Principles of Data Structures, Harvesting and Wrangling

Dr. Schenita Floyd

March 3rd, 2024

Project Overview

This project aims to understand the issue of lack of student engagement in group projects. Leveraging Google Cloud Platform (GCP), the team meticulously navigated the data lifecycle.

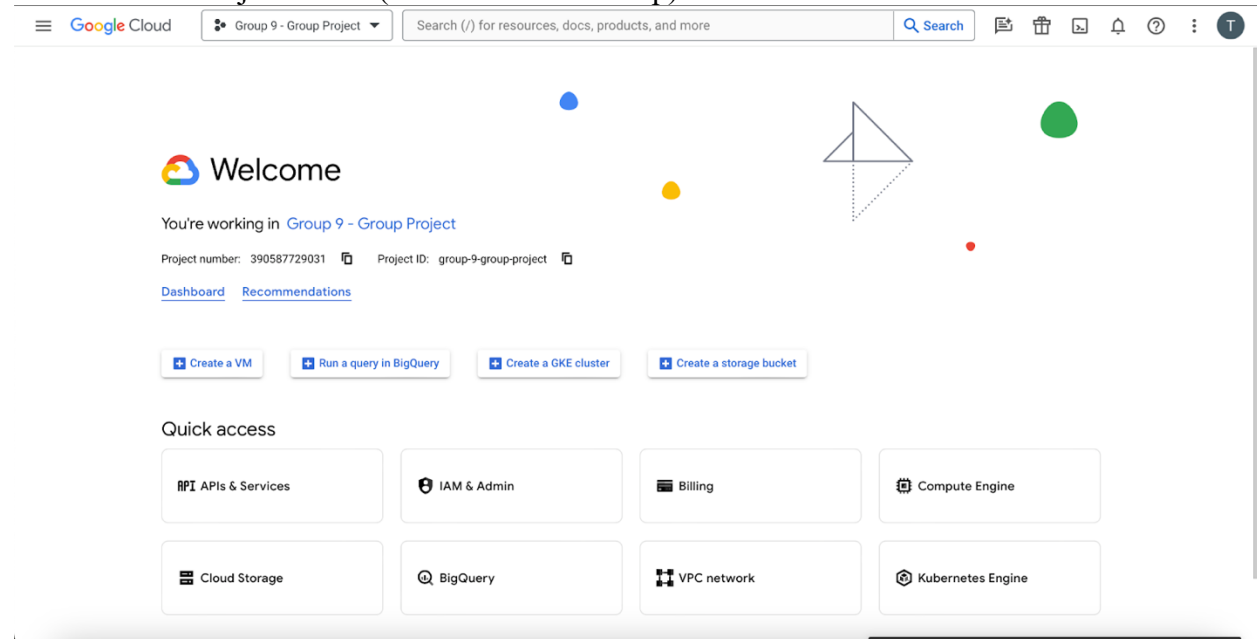
Initially, approved data sources from reputable platforms such as Data.gov and Reddit were identified and utilized. Data storage was established on GCP, with both static and streaming data sources uploaded for processing. Using GCP DataPrep by Trifacta, the team meticulously cleaned and preprocessed the data, ensuring its quality and relevance.

The processed data was then subjected to analysis using tools like Google Dataproc and BigQuery. Queries were executed to extract meaningful insights into student engagement patterns, shedding light on potential factors contributing to the lack of engagement.

Our Cloud Provider

Our team has utilized Google Cloud Platform (GCP) to address the lack of student engagement in group projects, leveraging various tools provided by GCP. We've established a project and set up a Hadoop infrastructure using Google Dataproc. Below are screenshots for your reference, showcasing the setup:

Screenshot 1: Project Name (Cloud Platform Setup)



Screenshot 2: Dataproc

Navigation menu

Jobs on Clusters

Clusters

Jobs

Workflows

Autoscaling policies

Serverless

Batches

Interactive

Metastore Services

Metastore

Federation

Utilities

Component exchange

Release Notes

Create a Dataproc cluster on Compute Engine

Set up cluster

Begin by providing basic information.

Configure nodes (optional)

Change node compute and storage capabilities.

Customize cluster (optional)

Add cluster properties, features, and actions.

Manage security (optional)

Change access, encryption, and security settings.

CREATE

CANCEL

EQUIVALENT COMMAND LINE

Number of local SSDs

x 375GB

Local SSD Interface

Worker nodes

Each contains a YARN NodeManager and a HDFS DataNode. HDFS replication factor is 2.

General purpose

Compute optimized

Memory optimized

GPUs

Machine types for common workloads, optimized for cost and flexibility

Series

E2

CPU platform selection based on availability

Machine type

e2-standard-4 (4 vCPU, 2 core, 16 GB memory)

vCPU

4

Memory

16 GB

CPU PLATFORM AND GPU

Number of worker nodes

2

Primary disk size

128

GB

Primary disk type

Standard Persistent Disk

Number of local SSDs

x 375GB

Local SSD Interface

Screenshot 3: Dataproc

Dataproc

Jobs on Clusters

Clusters

Jobs

Workflows

Autoscaling policies

Serverless

Batches

Interactive

Metastore Services

Metastore

Federation

Utilities

Component exchange

Release Notes

Clusters

CREATE CLUSTER

REFRESH

START

STOP

DELETE

REGIONS

+ 5 RECOMMENDED ALERTS

HIDE INFO PANEL

LE

Filter

Search clusters, press Enter

	Name	Status	Region	Zone	Total worker nodes	Flexible VMs?	Sched
<input type="checkbox"/>	dp-hadoop-spark-2-cluster-group9	Running	us-central1	us-central1-a	2	No	Off

Request to create cluster dp-hadoop-spark-2-cluster-group9 submitted

No clusters selected

PERMISSIONS

LABELS

Please select at least one resource.

Our Data Storage

We've established a dedicated storage area to manage our data sources. Our efforts include acquiring and uploading two types of data: one static file containing information on Chicago schools and one streaming data source from Reddit. Attached are screenshots showcasing our storage location and the uploaded data.

Screenshot 4: Dataset 1 in GCP Storage Bucket

The screenshot shows the Google Cloud Storage Buckets page. The left sidebar contains navigation links: Cloud Storage, Buckets, Monitoring, and Settings. The main content area displays a list of buckets. A table lists the bucket 'lackofengagement_data' with the following details:

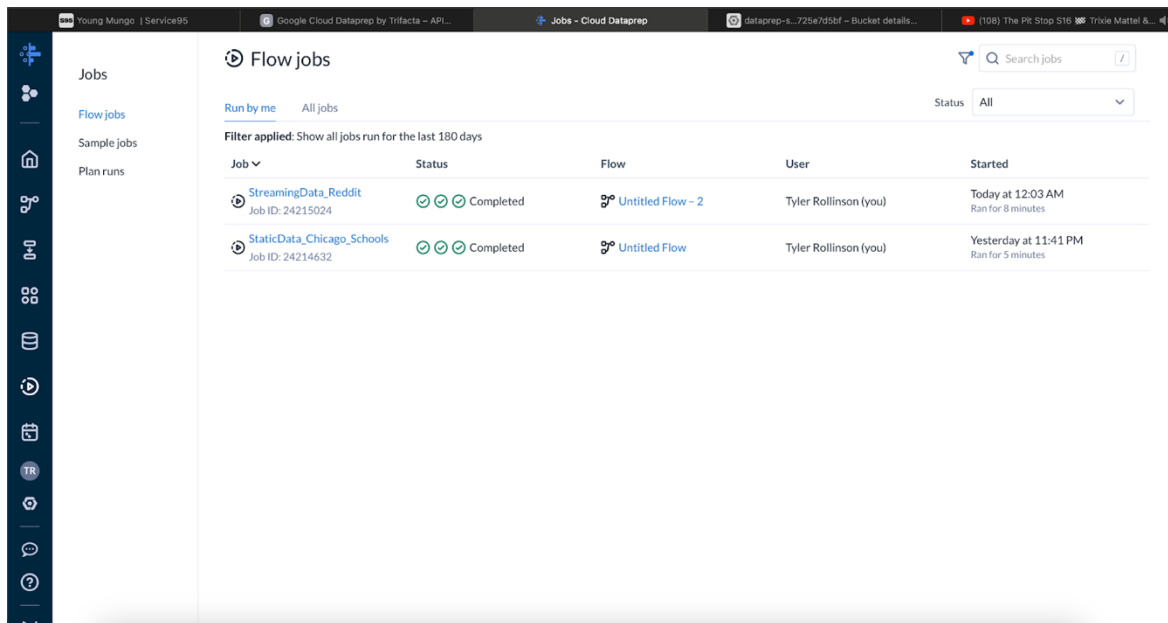
Name	Created	Location type	Location	Default storage class	Last modified	Public access
lackofengagement_data	Mar 5, 2024, 10:25:49 PM	Region	us-south1	Standard	Mar 5, 2024, 10:25:49 PM	Not public

Screenshot 5: Dataset 2

The screenshot shows the details page for the 'lackofengagement_data' bucket. The page displays the bucket's location (us-south1 (Dallas)), storage class (Standard), public access (Not public), and protection (None). Below this, there are tabs for OBJECTS, CONFIGURATION, PERMISSIONS, PROTECTION, LIFECYCLE, OBSERVABILITY, and INVENTORY REPORTS. The OBJECTS tab is active, showing a list of objects:

Name	Size	Type	Created	Storage class	Last modified	Public access	Version history
StaticData_Chicago_Schools.csv	1.3 MB	text/csv	Mar 5, 2024, 10:28:47 PM	Standard	Mar 5, 2024, 10:28:47 PM	Not public	—
StreamingData_Reddit.csv	1.7 MB	text/csv	Mar 5, 2024, 10:28:48 PM	Standard	Mar 5, 2024, 10:28:48 PM	Not public	—

Screenshot 6: Folders (Data Cleaning and Pre processing)



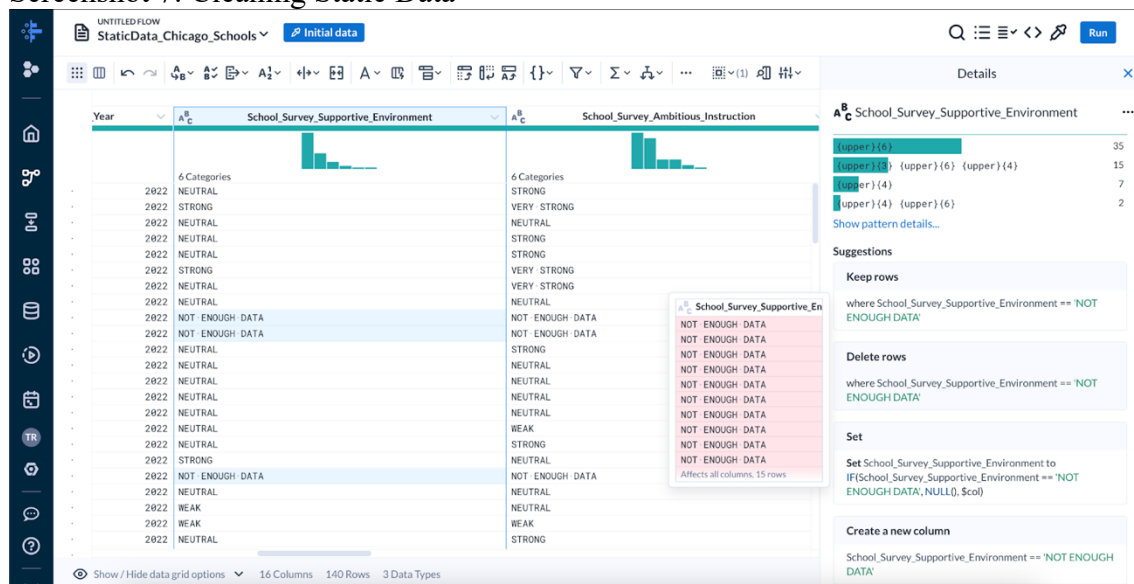
The screenshot shows the Google Cloud DataPrep interface. On the left is a sidebar with navigation icons. The main area is titled 'Flow jobs' and includes a search bar and a status filter set to 'All'. Below this, a table lists jobs with columns for Job, Status, Flow, User, and Started. Two jobs are shown, both completed by Tyler Rollinson.

Job	Status	Flow	User	Started
StreamingData_Reddit Job ID: 24215024	Completed	Untitled Flow - 2	Tyler Rollinson (you)	Today at 12:03 AM Ran for 8 minutes
StaticData_Chicago_Schools Job ID: 24214632	Completed	Untitled Flow	Tyler Rollinson (you)	Yesterday at 11:41 PM Ran for 5 minutes

Our Initial Tools

We employed GCP DataPrep by Trifacta to preprocess our data. The screenshots below illustrate the cleaning procedures applied to both the Chicago schools file and the Reddit file. Our initial data processing steps are detailed in the provided screenshots.

Screenshot 7: Cleaning Static Data



The screenshot shows the Google Cloud DataPrep interface for a job named 'StaticData_Chicago_Schools'. The main view displays a data grid with columns for Year, School_Survey_Supportive_Environment, and School_Survey_Ambitious_Instruction. A tooltip for 'School_Survey_Supportive_Environment' shows a list of values including 'NEUTRAL', 'STRONG', 'VERY STRONG', and 'NOT ENOUGH DATA'. On the right, a 'Details' panel shows a histogram for 'School_Survey_Supportive_Environment' with a pattern of {upper}(6), {upper}(4), {upper}(6), {upper}(4). Below the histogram, there are suggestions for 'Keep rows', 'Delete rows', 'Set', and 'Create a new column'.

Year	School_Survey_Supportive_Environment	School_Survey_Ambitious_Instruction
2022	NEUTRAL	STRONG
2022	STRONG	VERY STRONG
2022	NEUTRAL	NEUTRAL
2022	NEUTRAL	STRONG
2022	STRONG	VERY STRONG
2022	NEUTRAL	VERY STRONG
2022	NEUTRAL	NEUTRAL
2022	NOT ENOUGH DATA	NOT ENOUGH DATA
2022	NOT ENOUGH DATA	NOT ENOUGH DATA
2022	NEUTRAL	STRONG
2022	NEUTRAL	NEUTRAL
2022	NEUTRAL	NEUTRAL
2022	NEUTRAL	NEUTRAL
2022	NEUTRAL	NEUTRAL
2022	NEUTRAL	NEUTRAL
2022	STRONG	STRONG
2022	NOT ENOUGH DATA	NOT ENOUGH DATA
2022	NOT ENOUGH DATA	NEUTRAL
2022	WEAK	NEUTRAL
2022	WEAK	WEAK
2022	NEUTRAL	STRONG

Removed rows with missing data for graduation and college enrollment rates.

Screenshot 8: Cleaning Streaming Data

The screenshot shows the Google Cloud DataPrep interface for a project named 'StreamingData_Reddit'. The main view displays a list of comments from Reddit, each with a sentiment label. A 'Replace cells' dialog is open, showing a search for 'The sentiment expressed in the text is primarily negative.' and a replacement with 'Negative'. The dialog also shows a list of categories for the sentiment labels: Negative, Positive, and Mixed sentiment - Negative. The 'Replace with' field is set to 'Negative'.

Standardized data field for the sentiment_label to three data points; Negative, Positive, and Neutral. Removed all rows of data that had missing data in the “text” field.

After completing the data cleaning process, we uploaded the files and executed two queries. Attached are screenshots depicting the queries we performed.

Screenshot 9: Big Query Example 1

The screenshot shows the Google Cloud BigQuery interface. The query editor displays a query that selects data from 'School_Survey_Supportive_Environment' and 'Avg_Graduation_4_Year_Rate'. The query results are displayed in a table with columns: Row, School_Survey_Supportive_Environment, Avg_Graduation_4_Year_Rate, and Avg_College_Enrollment_Rate. The results show a correlation between supportive environment and graduation rate.

Row	School_Survey_Supportive_Environment	Avg_Graduation_4_Year_Rate	Avg_College_Enrollment_Rate
1	WEAK	75.7	49.216666666666666
2	VERY WEAK	71.6	35.3
3	VERY STRONG	95.94999999999999	90.75
4	STRONG	80.30857142857143	59.23999999999999
5	NEUTRAL	74.39873417721518	52.246153846153846

We ran a query to see if there is any correlation between a schools “supportive enviroment” rank and the graduation or college enrollment rate. The data shows that there is a strong correlation between a higher (Very Strong) supportive ranking and a higher graduation and college enrollment rate.

Screenshot 10: BigQuery example 2

The screenshot displays the Google Cloud BigQuery interface. The top navigation bar includes the Google Cloud logo, a project selector set to 'Group-9-Group-Project', a search bar, and various utility icons. The left sidebar contains a navigation menu with categories like Analysis, Migration, and Administration, with 'BigQuery Studio' selected under Analysis.

The main workspace is divided into three sections:

- Explorer:** A sidebar on the left showing a tree view of resources. Under 'StaticData', 'ChicagoSchools' and 'ImpactAnalysis' are visible. Under 'StreamingData', 'Reddit Sentiments' is listed and selected.
- Query Editor:** The central area shows a SQL query for 'Untitled 3'. The query uses a CTE named 'SentimentCounts' to calculate the total count of posts and the average upvotes for each sentiment label (Negative, Neutral, Positive) from the 'Reddit Sentiments' table.
- Query results:** The bottom section displays the results of the query in a table format. It includes tabs for 'JOB INFORMATION', 'RESULTS' (selected), 'CHART', 'JSON', 'EXECUTION DETAILS', and 'EXECUTION GRAPH'.

The 'Query results' table shows the following data:

Row	sentiment_label	TotalPosts	AverageUpvotes	WeightedAverageUpvotes
1	Negative	810	84.62839506172...	70.81508264462...
2	Neutral	67	8.223880597014...	0.569214876033...
3	Positive	91	87.91208791208...	8.264462809917...

We ran a query on the data from Reddit to understand the popularity of sentiments towards group projects. We first retrieved a total count of all sentiment ratings in the dataset. We then calculated the average upvotes for each rating; Negative, Positive, Neutral. Given that the dataset was largely populated with “Negative” sentiments, we then weighed the average upvote to more fairly represent the popularity of opinion. The data shows that from our data set “Negative” sentiments are 8x’s more frequent than positive sentiments.

Screenshot 11: Hive Query

```
ssh.cloud.google.com
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

INFO : Completed compiling command(queryId=hive_20240307210719_330d71b0-80f0-4405-a71e-73e2fa2b796c); Time taken: 0.206 seconds
INFO : Concurrency mode is disabled, not creating a lock manager
INFO : Executing command(queryId=hive_20240307210719_330d71b0-80f0-4405-a71e-73e2fa2b796c): SELECT sentiment_label, COUNT(*) AS count
FROM sentiment_analysis
GROUP BY sentiment_label
WARN : Hive-on-MR is deprecated in Hive 2 and may not be available in the future versions. Consider using a different execution engine (i.e. spark, tez) or using Hive 1.X releases.
INFO : Query ID = hive_20240307210719_330d71b0-80f0-4405-a71e-73e2fa2b796c
INFO : Total jobs = 1
INFO : Launching Job 1 out of 1
INFO : Starting task (Stage-1:MAPRED) in serial mode
INFO : Number of reduce tasks not specified. Estimated from input data size: 1
INFO : In order to change the average load for a reducer (in bytes):
INFO :   set hive.exec.reducers.bytes.per.reducer=<number>
INFO : In order to limit the maximum number of reducers:
INFO :   set hive.exec.reducers.max=<number>
INFO : In order to set a constant number of reducers:
INFO :   set mapreduce.job.reduces=<number>
INFO : number of splits:1
INFO : Submitting tokens for job: job_1709844499804_0004
INFO : Executing with tokens: []
INFO : The url to track the job: http://dp-hadoop-spark-2-cluster-group9-m:8088/proxy/application_1709844499804_0004/
INFO : Starting Job = job_1709844499804_0004; Tracking URL = http://dp-hadoop-spark-2-cluster-group9-m:8088/proxy/application_1709844499804_0004/
INFO : Kill Command = /usr/lib/hadoop/bin/mapred job -kill job_1709844499804_0004
INFO : Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 1
INFO : 2024-03-07 21:07:32,189 Stage-1 map = 0%, reduce = 0%
INFO : 2024-03-07 21:07:40,450 Stage-1 map = 100%, reduce = 0%, Cumulative CPU 3.98 sec
INFO : 2024-03-07 21:07:49,770 Stage-1 map = 100%, reduce = 100%, Cumulative CPU 8.09 sec
INFO : MapReduce Total cumulative CPU time: 8 seconds 90 msec
INFO : Ended Job = job_1709844499804_0004
INFO : MapReduce Jobs Launched:
INFO : Stage-Stage-1: Map: 1 Reduce: 1 Cumulative CPU: 8.09 sec HDFS Read: 26382 HDFS Write: 189 SUCCESS
INFO : Total MapReduce CPU Time Spent: 8 seconds 90 msec
INFO : Completed executing command(queryId=hive_20240307210719_330d71b0-80f0-4405-a71e-73e2fa2b796c); Time taken: 31.698 seconds
INFO : OK
INFO : Concurrency mode is disabled, not creating a lock manager

+-----+-----+
| sentiment_label | count |
+-----+-----+
| Negative        | 810   |
| Neutral         | 67    |
| Positive        | 91    |
| sentiment_label | 1     |
+-----+-----+
4 rows selected (31.959 seconds)
0: jdbc:hive2://localhost:10000
```

Screenshot 12: Spark Query

```
ssh.cloud.google.com
SSH-in-browser
UPLOAD FILE
DOWNLOAD FILE

the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Mar 7 20:50:01 2024 from 35.235.244.33
tyler_rolinson@dp-hadoop-spark-2-cluster-group9-m:~$ spark-sql
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
ivysettings.xml file not found in HIVE HOME or HIVE CONF DIR,/etc/hive/conf.dist/ivysettings.xml will be used
24/03/07 21:10:28 INFO org.apache.spark.SparkEnv: Registering MapOutputTracker
24/03/07 21:10:28 INFO org.apache.spark.SparkEnv: Registering BlockManagerMaster
24/03/07 21:10:28 INFO org.apache.spark.SparkEnv: Registering BlockManagerMasterHeartbeat
24/03/07 21:10:28 INFO org.apache.spark.SparkEnv: Registering OutputCommitCoordinator
Spark master: yarn, Application Id: application_1709844499804_0005
spark-sql> show tables;
default reddit sentiment      false
default reddit sentiments     false
default sentiment_analysis     false
default sentiment_data        false
Time taken: 4.31 seconds, Fetched 4 row(s)
spark-sql> SELECT * FROM sentiment_analysis LIMIT 5;
24/03/07 21:11:18 WARN org.apache.hadoop.hive.gl.session.SessionState: METASTORE_FILTER_HOOK will be ignored, since hive.security.authorization.manager
is set to instance of HiveAuthorizerFactory.
24/03/07 21:11:19 WARN org.apache.hadoop.util.concurrent.ExecutorHelper: Thread (Thread[GetFileInfo #1,5,main]) interrupted:
java.lang.InterruptedException
  at com.google.common.util.concurrent.AbstractFuture.get (AbstractFuture.java:510)
  at com.google.common.util.concurrent.AbstractFuture$TrustedFuture.get (AbstractFuture.java:88)
  at org.apache.hadoop.util.concurrent.ExecutorHelper.logThrowableFromAfterExecute (ExecutorHelper.java:48)
  at org.apache.hadoop.util.concurrent.HadoopThreadPoolExecutor.afterExecute (HadoopThreadPoolExecutor.java:90)
  at java.util.concurrent.ThreadPoolExecutor.runWorker (ThreadPoolExecutor.java:1157)
  at java.util.concurrent.ThreadPoolExecutor$Worker.run (ThreadPoolExecutor.java:624)
  at java.lang.Thread.run (Thread.java:750)
NULL
sentiment_label
667 Negative
667 Negative
3 Negative
1 Negative
Time taken: 6.105 seconds, Fetched 5 row(s)
spark-sql> SELECT sentiment_label, COUNT(*) AS count
> FROM sentiment_analysis
> GROUP BY sentiment_label;
sentiment_label 1
Neutral 67
Positive 91
Negative 810
Time taken: 3.073 seconds, Fetched 4 row(s)
spark-sql>
```


Here is the time summary test comparison between hive and spark:

Hive Query Times:

Query 1: SELECT * FROM sentiment_analysis LIMIT 5; Time: 24.685 seconds

Query 2: SELECT sentiment_label, COUNT(*) AS count FROM sentiment_analysis
GROUP BY sentiment_label; Time: 31.959 seconds

Spark SQL Query Times:

Query 1: SELECT * FROM sentiment_analysis LIMIT 5; Time: 6.105 seconds

Query 2: SELECT sentiment_label, COUNT(*) AS count FROM sentiment_analysis
GROUP BY sentiment_label; Time: 3.073 seconds

Supporting Documents

Chicago Schools Survey

Dates: 03/01/2024

Website: <https://catalog.data.gov/dataset/chicago-public-schools-school-progress-reports-sy2324>

Quality: Data is in a consistent format but will employ a more abstract approach to our research as the data does not directly reference student engagement in group projects. We found this to be difficult subject to research with limited data sources available to us. We will be comparing schools “supportive environment” ranking with the graduation rate to understand if there is any correlation between the two.

Reddit API “Group Projects”

Subreddits: r/college; r/education; r/StudentLife

Dates: 03/05/2024

Website: www.reddit.com/dev/api/

Quality: Data is more relevant to our research topic, however, there are numerous fields with missing data and inconsistent data format. Data requires more transformation to be useful in analysis.