Pulse of Data Science on Reddit

Srikar Devulapalli (vdevula@iu.edu)

Introduction

In this digital age, the landscape of knowledge exchange is vast, and platforms like Reddit are at the forefront of community-based learning, especially in rapidly evolving fields like Data Science.

This hands-on project report showcases a data pipeline that captures and analyzes the collective intelligence of data science communities on Reddit by dissecting over 500k posts from various data science subreddits to distill the essence of community discussions, identify the most engaging topics, and explore the dynamics of post popularity.

Background

For this project, I chose to look at Reddit's data science posts because Reddit is where a lot of people who like data science go to talk about it. People ask questions, share information, and talk about new things happening in data science there.

The good thing about using Reddit is that there are a lot of posts to look at. This means we can really understand the topics that people who are interested in data science are talking about and what they think is important. As we add more posts to our study—up to 750,000 or even a million—we'll be able to see better what topics are getting more popular over time.

We want to figure out which topics get people excited and which ones get the most attention. This could help us see what will become more important in data science in the future. The goal is to give a clear

picture of what's happening right now in the world of data science on Reddit.

Methodology

Utilizing a rich dataset drawn from numerous data science subreddits, I had established a pipeline that begins with data extraction from Reddit, employs a distributed file system for data handling and processing, and leverages the cloud capabilities of Google Cloud Platform (GCP) to store and analyze the data at scale.

By employing tools designed for handling big data, such as Apache Spark within GCP's Dataproc service, I had ensured that our approach is not only robust but also scalable and reproducible.

I started building my pipeline by creating a new project in GCP as shown in Fig.1

Fig.1: Project Creation in GCP

Fig.2 shows an overview of data pipeline created and the steps involved.

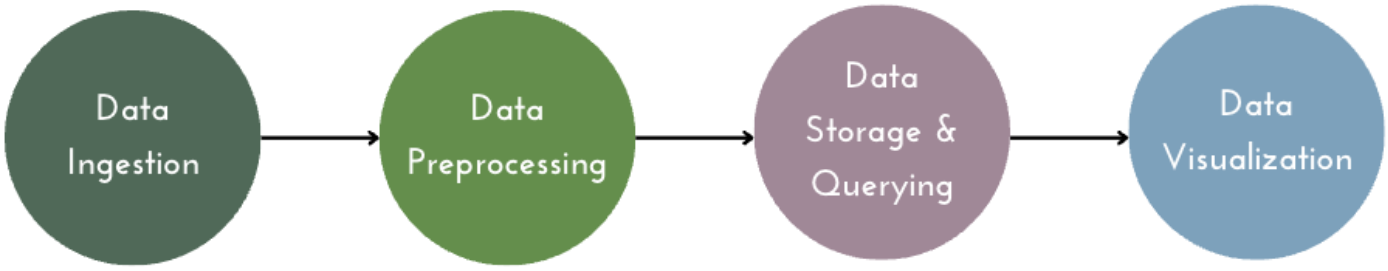

Fig.2: Data pipeline

Here's a detailed explanation of how the data pipeline is built and what was achieved.

First, I had started to enable the Cloud Functions, Cloud Scheduler, Cloud Storage, Big Query and Secret Manager APIs required for this project (Fig.3).

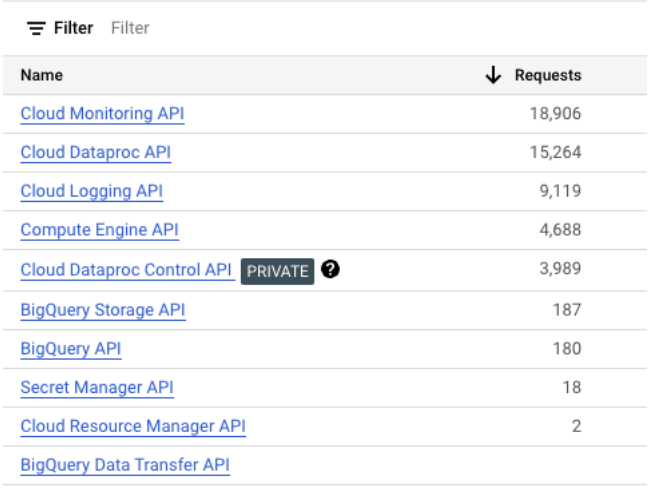

Fig.3: Enabling required APIs

1. **Data Ingestion:**

The data is located in Kaggle: https://www.kaggle.com/datasets/maksymshkliarevskyi/reddit-data-science-posts/data

It contains 500k+ rows across 13 columns, with a size of 340.11 MB.

On creating a GCP bucket named ‘reddit-ds-bucket’, I was able to run a script in Kaggle notebook to import data to this bucket.

Script for Data Ingestion:

```
import os
from google.cloud import storage

storage_client = storage.Client(project='fa23-i535-vdevula-redditdspost') #set the ID of your project at GCP

#function for uploading the data from 4aggle to google cloud services

def upload_files(bucket_name, source_folder):
    bucket = storage_client.get_bucket(bucket_name)
    for filename in os.listdir(source_folder):
        blob = bucket.blob(filename)
        blob.upload_from_filename(source_folder + filename)

local_data ='/4aggle/input/reddit-data-science-posts/'
upload_files('reddit-ds-bucket', local_data) #calling the dunction for uploading the data from 4aggle to google cloud services
```

2. Data Pre-processing:

In the Data Pre-processing phase, I had chosen to use Dataproc cluster’s distributed file system computing because it’s specially designed to handle and process big data effectively. This system spreads the data across multiple servers, allowing us to work on different parts of the data at the same time.

I created a Dataproc cluster named ‘reddit-ds-cluster’ with the image version able to run Spark jobs and machine type was ‘e2-standard-4’, which I thought would be apt for this dataset (Fig.4)

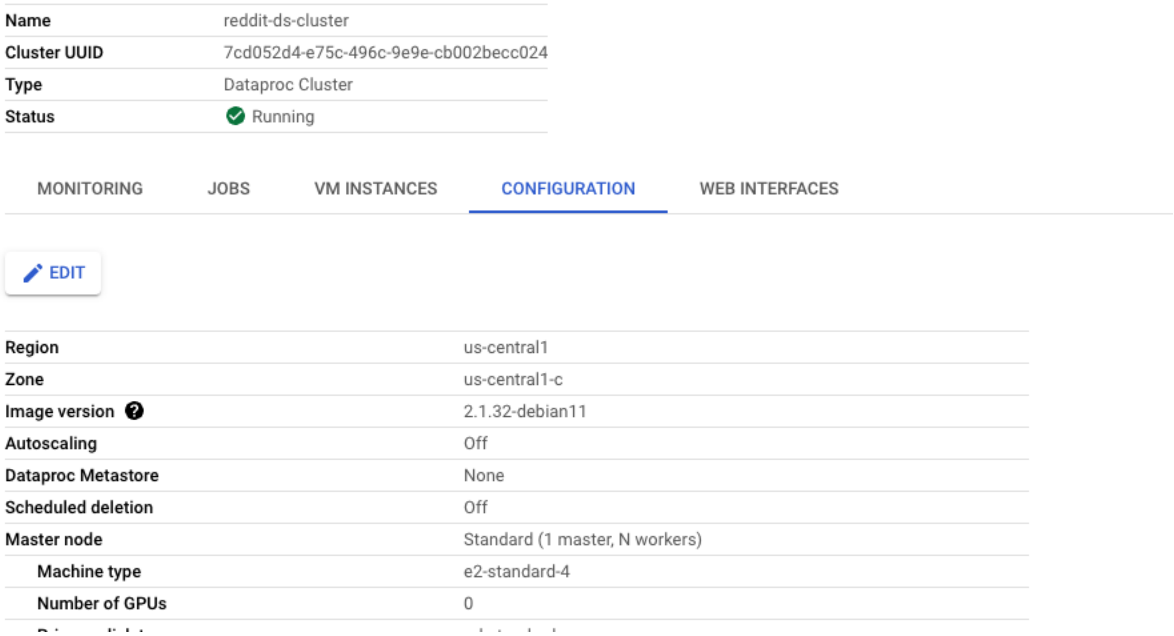

Fig.4 : Dataproc Cluster

Later, with the master VM Instance present in this cluster, I was able to implement a distributed file system, where I utilized Apache Spark job run on the Dataproc cluster.
Apache Spark is renowned for its ability to process large datasets quickly and efficiently. When I ran Spark jobs within the cluster, they utilized the distributed nature of Dataproc’s file system, where data is split across multiple nodes.
This system spreads the data across multiple servers, allowing us to work on different parts of the data at the same time. It’s much faster than trying to handle all that data on a single machine and is essential when dealing with large datasets like the ones from Reddit.

Here's the script that I initially saved as 'pyspark-script.py' in GCS bucket and ran it on SSH of the master VM instance in Dataproc using the command in Fig.5 and Fig.6

```
vdevula@reddit-ds-cluster-m:~$ gcloud dataproc jobs submit
 pyspark --cluster=reddit-ds-cluster --region=us-central1
gs://reddit-ds-bucket/pyspark-script.py -- gs://reddit-ds-
bucket/reddit_database.csv
```

Fig.5: glcoud command to run Pyspark job

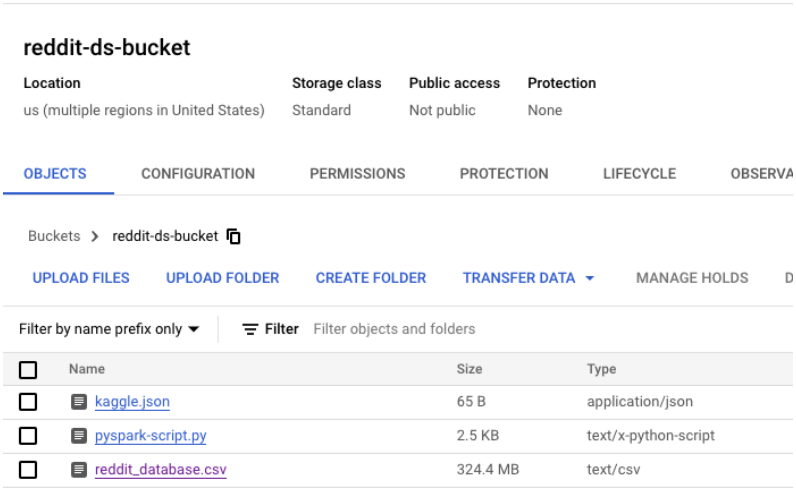

Fig.6: Storage in GCS bucket

I performed the following pre-processing and data manipulation steps to achieve the project objectives:

- In order to calculate the posts across sub-reddit, I created a data table named 'num_posts_per_subreddit'
- Similar to the previous operation, I calculated average score across subreddit and stored in the table named 'avg_score_per_subreddit'
- To understand the trending topics in data science, I decided to build a word cloud for which I removed null or empty rows from the column title and did explode operation, which fragments text

in this column to individual words and found the frequency of each word and stored this data in a table named ‘title_words’

pyspark-script.py

```
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, explode, split, lower, avg, desc, regexp_replace, count, size
from pyspark.ml.feature import StopWordsRemover
from pyspark.sql.types import StringType, IntegerType, FloatType, ArrayType, StructType,StructField

# Initialize Spark session with BigQuery support
spark = SparkSession.builder \
    .appName("RedditDataAnalysis") \
    .config('spark.jars.packages', 'com.google.cloud.spark:spark-bigquery-with-dependencies_2.12:0.23.2') \
    .getOrCreate()

# Define the path to the data
data_path = 'gs://reddit-ds-bucket/reddit_database.csv'

# Read data from GCS into DataFrame with the defined schema
reddit_data = spark.read.csv(data_path, header=True, inferSchema=True)

# Remove subreddits with more than two words
reddit_data = reddit_data.filter(size(split(col("subreddit"), " ")) <= 2)

# 1. Number of posts per subreddit
num_posts_per_subreddit = reddit_data.groupBy('subreddit').count()
num_posts_per_subreddit.write.format('bigquery') \
    .option("writeMethod", "direct") \
    .option('table', 'fa23-i535-vdevula-redditdspost.reddit_ds_dataset.num_posts_per_subreddit') \
    .mode('overwrite') \
    .save()

# 2. Top words for word cloud based on title column
# Filter out rows with null or empty title
reddit_data = reddit_data.filter(col('title').isNotNull() & (col('title') != ""))
```

```
# Normalize the title text and tokenize
reddit_data = reddit_data.withColumn('words', split(lower(col('title')), "\\s+"))
# Remove stop words
remover = StopWordsRemover(inputCol='words', outputCol='filtered_words')
reddit_data = remover.transform(reddit_data)
# Explode the filtered words into individual words and count
title_words = reddit_data.withColumn('word', explode(col('filtered_words'))) \
                    .select('word') \
                    .groupBy('word') \
                    .count() \
                    .orderBy(desc('count')) \
                    .limit(50)
title_words.write.format('bigquery') \
    .option("writeMethod", "direct") \
    .option('table', 'fa23-i535-vdevula-redditdspost.reddit_ds_dataset.title_words') \
    .mode('overwrite') \
    .save()

# 3. Average score based on the subreddit column
avg_score_per_subreddit = reddit_data.groupBy('subreddit').agg(avg('score').alias('avg_score'))
avg_score_per_subreddit.write.format('bigquery') \
    .option("writeMethod", "direct") \
    .option('table', 'fa23-i535-vdevula-redditdspost.reddit_ds_dataset.avg_score_per_subreddit') \
    .mode('overwrite') \
    .save()

# Stop Spark session
spark.stop()
```

Further, job monitoring was done to ensure the data tables were stored in Big Query using Jobs and Monitoring Dashboard in Cluster as shown in Fig.7

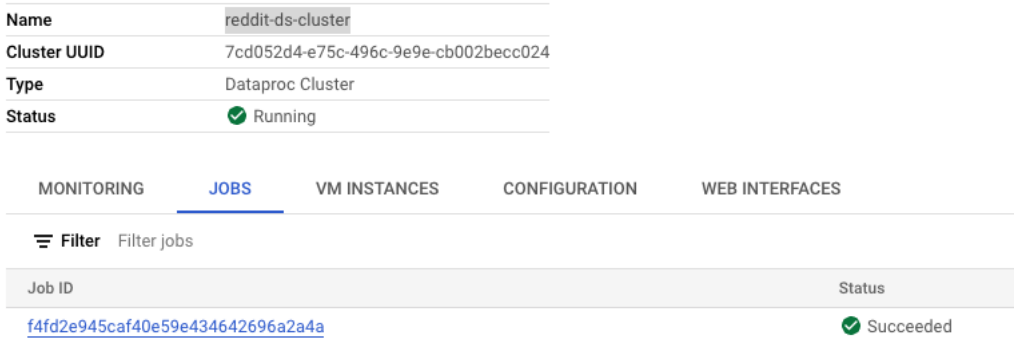

Fig.7: Job monitoring

3. Data Storage & Querying

After the Data tables were prepared in the Big Query (Fig.8), I had procceded to query the required results and make them available for visualization.

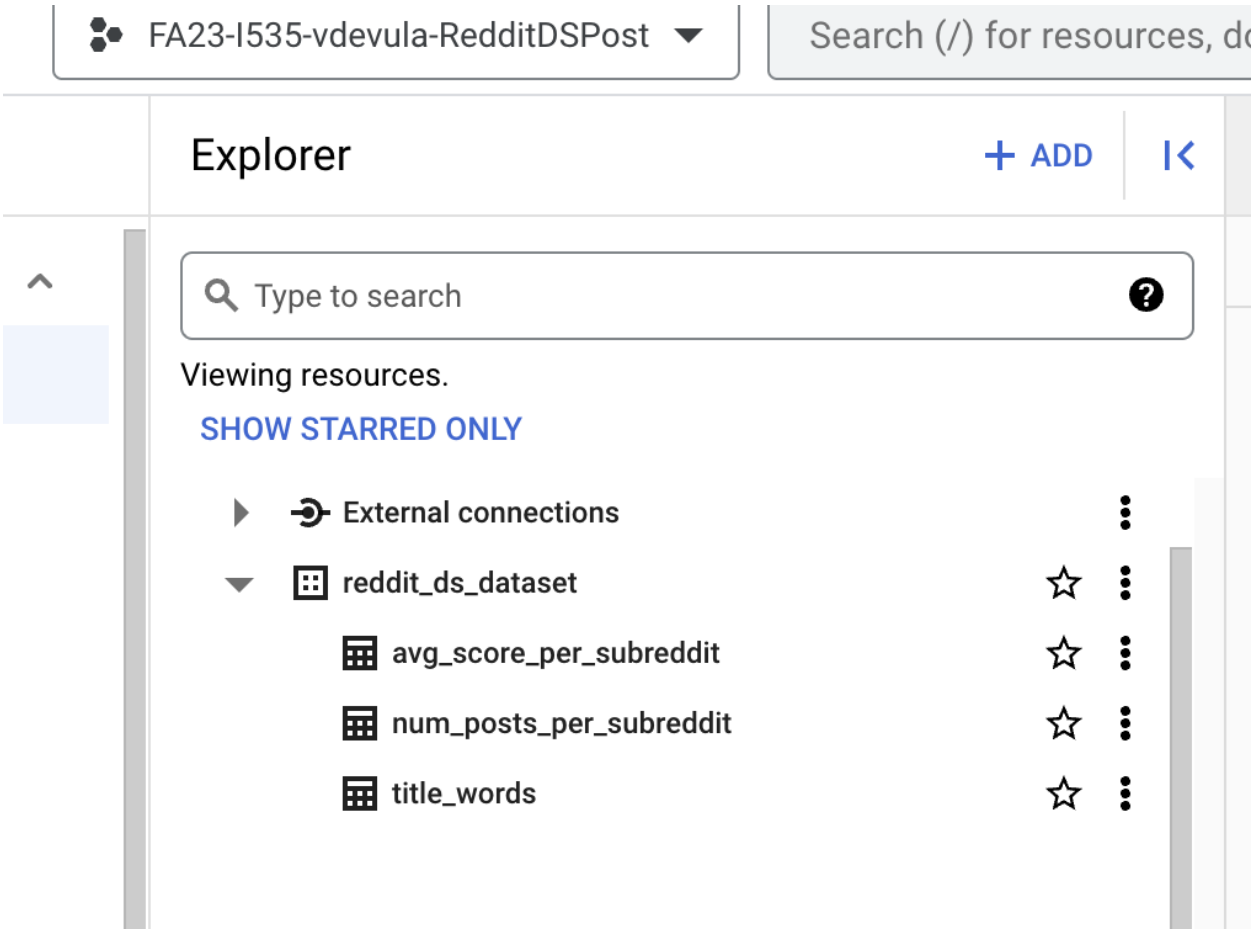

Fig.8: Tables available in Big Query

Here are the sample queries that I have created:

- Top 10 subreddits based on the posts that are linked to

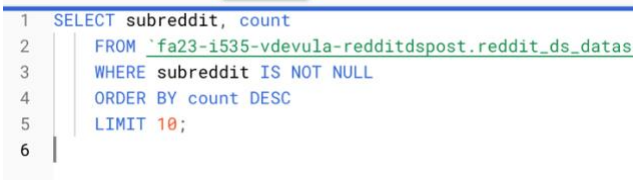

Fig.9.a: Top 10 subreddits by posts

- Top subreddits by avgscore

```
SELECT subreddit, avg_score
    FROM `fa23-i535-vdevula-redditdspost.reddit_ds_dataset.avg_score_per_subreddit`
    WHERE avg_score IS NOT NULL AND subreddit != ' probability'
    ORDER BY avg_score DESC
    LIMIT 20;
```

Fig.9.b: Top 20 subreddits by average score

- Word cloud data

```
SELECT *
    FROM `fa23-i535-vdevula-redditdspost.reddit_ds_dataset.title_words`
    ORDER BY 2
    LIMIT 50;
```

Fig.9.c: Top 50 words in title of a reddit post

4. Data Visualization

After creating these queries, I utilized the option to ‘Explore with Python notebook’ under Explore data Section in Query results tab

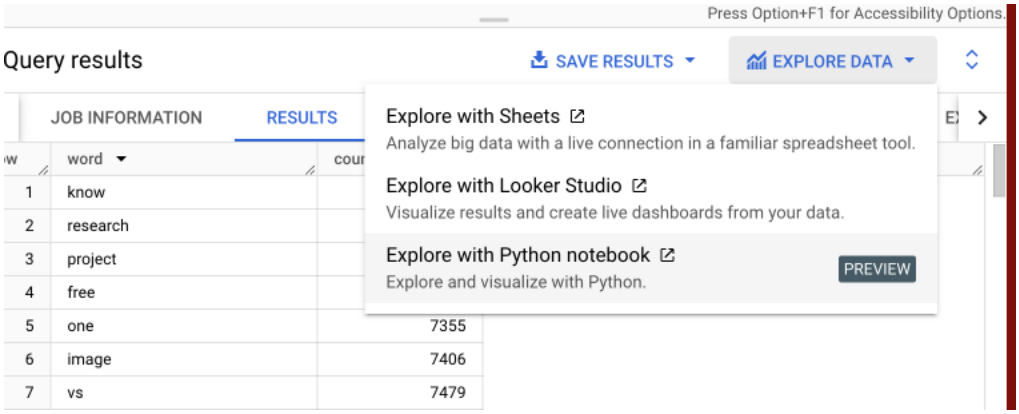

Fig.10: Option to visualize data

I have utilized this code snippet to generate results:

```
# @title Setup
from google.colab import auth
from google.cloud import bigquery
from google.colab import data_table

project = 'fa23-i535-vdevula-redditdspost' # Project ID inserted based on the query results selected to explore
location = 'US' # Location inserted based on the query results selected to explore
client = bigquery.Client(project=project, location=location)
data_table.enable_dataframe_formatter()
auth.authenticate_user()

"""Word Cloud"""

job = client.get_job('bquxjob_38866226_18bf969665c') # Job ID inserted based on the query results selected to explore
results = job.to_dataframe()

import pandas as pd
import matplotlib.pyplot as plt
```

```
from wordcloud import WordCloud
words_df = results

# Generate a word cloud
wordcloud = WordCloud(width = 800, height = 400, background_color
='white').generate_from_frequencies(dict(zip(words_df.word, words_df['count'])))
plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)
plt.show()

"""Top 10 Subreddits by Number of Posts"""

job = client.get_job('bquxjob_3af8681d_18bf984d2a1') # Job ID inserted based on the query results selected to
explore
results = job.to_dataframe()

# Plot the results as a bar chart
plt.figure(figsize=(12, 6))
plt.bar(results['subreddit'], results['count'], color='skyblue')
plt.xlabel('Subreddit')
plt.ylabel('Number of Posts')
plt.title('Top 10 Subreddits by Number of Posts')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()  # Adjust the padding between and around subplots

# Display the plot
plt.show()

job = client.get_job('bquxjob_7a1607d_18bf993e65d') # Job ID inserted based on the query results selected to
explore
results = job.to_dataframe()

# Plot the results as a bar chart
plt.figure(figsize=(12, 6))
plt.bar(results['subreddit'], results['avg_score'], color='skyblue')
```

```
plt.xlabel('Subreddit')
plt.ylabel('Average Score')
plt.title('Top 20 Subreddits by Average score')
plt.xticks(rotation=45, ha='right')
plt.tight_layout()  # Adjust the padding between and around subplots

# Display the plot
plt.show()
```

Cleaning up was also another underrated but crucial step in this process. As we achieved our goals and accomplished the tasks, it is important to clean up the resources used, making it beneficial for reducing unwanted organizational costs. So, I had deleted every resource after usage.

Results

Results w.r.t computation

This pipeline successfully worked from end-to-end without any discrepancies. The jobs were running fine, the data was populating and the visualizations were coming up.

Results w.r.t data

Coming to the data results, Fig. 11 shows the word cloud of the most frequently used terms in reddit posts related to Data Science.

- It can be observed that the terms ‘data’, ‘machine’ learning’ take the top 3 places

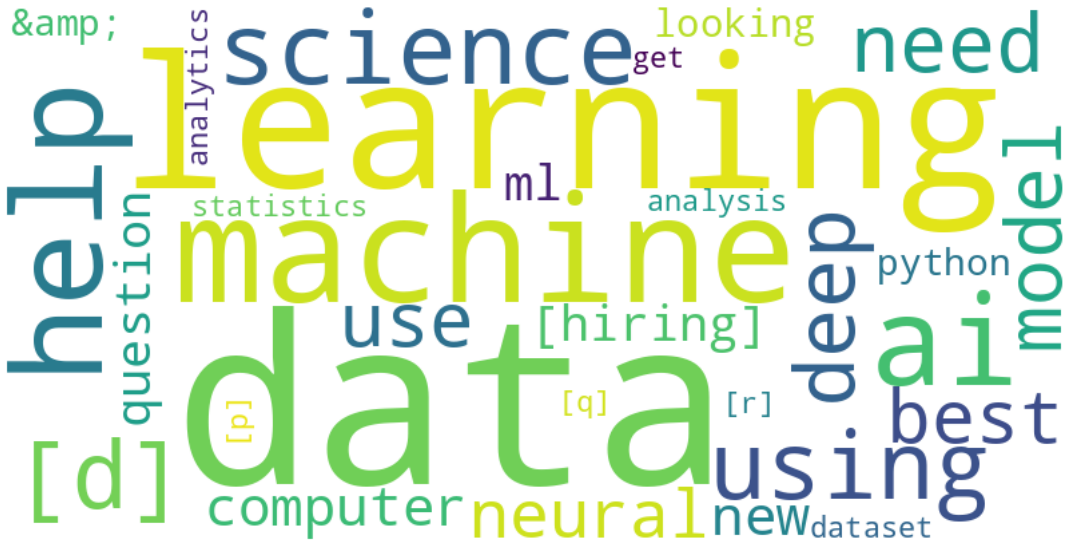

Fig.11: Word Cloud for Most Frequently repeated words in post title

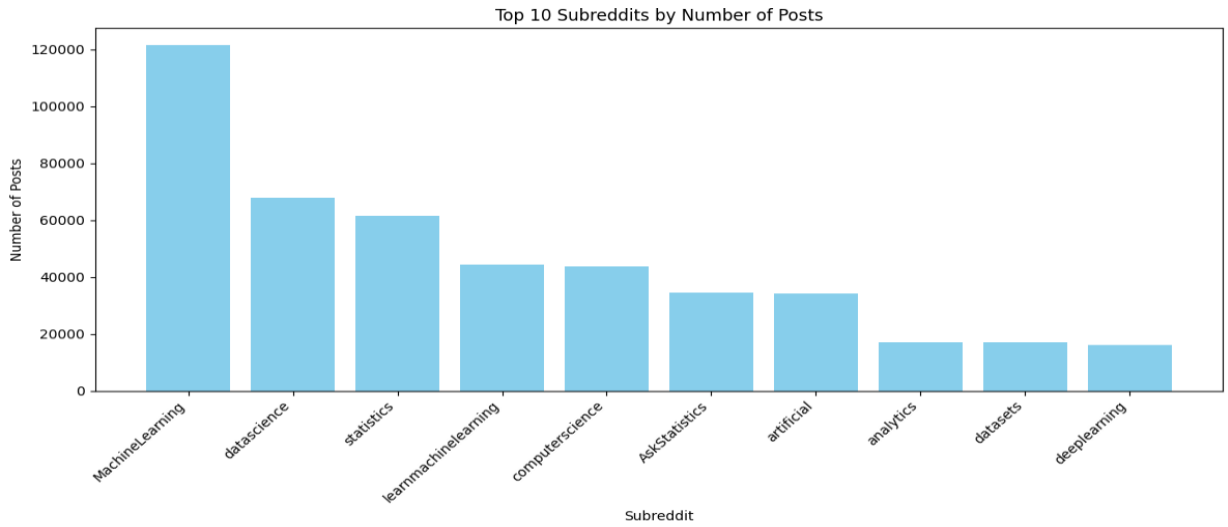

Fig.12: Top 10 Subreddits by Number of Posts

Fig. 12 shows the top 10 subreddits by number of posts.
Machine Learning, Data Science, Statistics, Computer Science, Deep Learning were among the uniquely repeated top subreddits

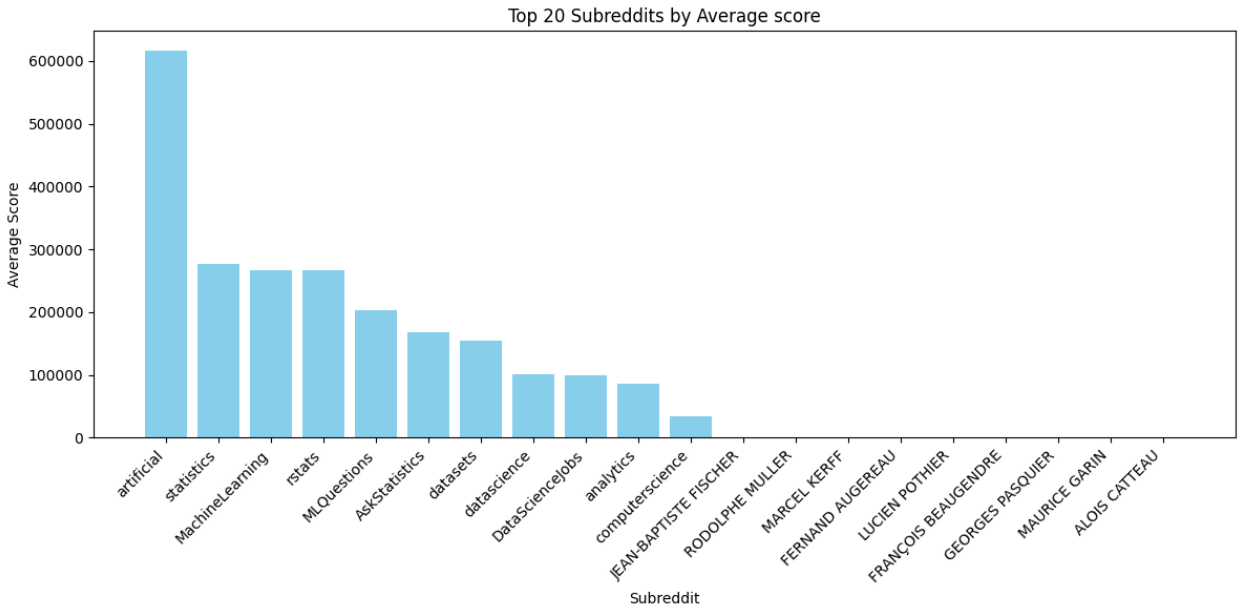

Fig.13: Top 20 Subreddits by Average Score

Fig.13 shows the top 10 subreddits by Average score.
'Artificial', 'statistics', 'machine learning', 'rstats', 'ML Questions' were among the top 5.

Discussion

Interpretation of results

- Machine learning and Python are at the heart of discussions, as seen in the word cloud. This concentration on machine learning is indicative of the field's status as a cornerstone of modern data science.
- In addition, from the word cloud we can see that post titles include 'question', 'help' are frequently posted in the community and this is pretty useful to the people in the community, which can also be backed by the fact that ML Questions have a top average score as seen in Fig.13

- Hiring related posts are also being posted on Reddit frequently, indicating the community's exposure to job market and opportunities which are useful for data science job aspirants.
- The bar chart in Fig.12 points to a hub of activity within a select few subreddits, with 'Machine Learning' and 'Data Science' standing out. This could reflect the community's inclination towards evolving technologies and methodologies.
- In Fig.13, the bar chart displays the top 20 subreddits by average score. Subreddits such as 'Artificial,' 'Statistics,' and 'Machine Learning' not only have high activity but also high scores, suggesting that their content is particularly well-received.
- 'Rstats' and 'ML Questions,' though not as active, still rank in the top five, indicating quality discussions that engage the community despite fewer posts.

Employing the skills/technologies from course

In my project, I tackled the challenge of analyzing a wealth of data science posts from Reddit. I began by transferring the data from Kaggle to the Google Cloud Platform, making sure it was all in one place and easy to work with. Using Apache Spark, a powerful tool I learned about during the course, I processed the data across a distributed file system. This meant I could clean and organize the data quickly because the work was spread out over many computers.

Once the data was ready, I stored it in BigQuery, Google's tool for looking through big datasets. I ran queries to pull out the information I needed, a process that I had practiced with similar technologies during my coursework.

Finally, I used visualization techniques to make sense of all this information in the pipeline. The goal was to show the most talked-about topics and trends in the data science community on Reddit in a way that was clear and easy to understand, using the principles I'd learned about presenting data effectively.

Barriers encountered

I was initially trying to call the Kaggle function in SSH of VM Instance to download data through Dataproc cluster into GCS bucket using this command multiple times after configuring and setting up Kaggle environment in gcloud:
kaggle datasets download -d maksymshkliarevskyi/reddit-data-science-posts
However, that didn't work. So, I ran a script in Kaggle notebook to transfer data to GCS bucket

Conclusion

In conclusion, my project successfully navigated the complexities of big data analysis, from the initial data ingestion from Kaggle to insightful visualizations using GCP's suite of tools.

This experience consolidated my understanding of distributed computing and cloud-based analytics, showcasing the power of these platforms in managing and interpreting large datasets.

Moving forward, I plan to refine the data pipeline for greater efficiency, explore more advanced visualization techniques, and expand the dataset to include additional sources, aiming to enrich the analysis and uncover deeper insights into the ever-evolving field of data science.

References:

- https://libinruan.gitlab.io/2021/01/15/kaggle101-Transfer-Competition-Data-into-Google-Buckets/
- https://www.cloudskillsboost.google/focuses/672?parent=catalog
- https://cloud.google.com/bigquery?hl=en
- https://www.analyticsvidhya.com/blog/2022/01/google-cloud-platform/
- https://cloud.google.com/dataproc/docs/tutorials/bigquery-connector-spark-example