Reddit Sentiment Analysis

Jimi Mehta (40225526) , Vidhi Sagathiya (40232374) , Anurag Agarwal (40232644) , Kalinga Swain (40226333) , and Muqaddaspreet Singh Bhatia (40276333)

Concordia University

Abstract

In the dynamic world of social media, Reddit stands out as a diverse platform hosting a myriad of opinions, discussions, and sentiments. To understand the nature of interactions and opinions expressed across various subreddits, a comprehensive sentiment analysis has been conducted using a dataset sourced from Kaggle. The dataset comprises posts from multiple subreddits, offering a rich tapestry of themes and emotions. Our objectives are multifaceted: to identify prevailing sentiments across different subreddits, to uncover any temporal or thematic patterns, and to understand the impact of various factors such as post time, subreddit theme, and user engagement on the overall sentiment. Key questions guiding our analysis include: What are the predominant emotions expressed in different subreddits? Is there any discernible pattern in sentiment across different times of the day or specific periods? How do subreddit themes influence the nature of discussions and sentiments? Through a blend of natural language processing techniques and sentiment analysis tools, this study not only aims to provide insights into the emotional undercurrents of Reddit but also seeks to contribute to the broader understanding of digital communication patterns and user behavior on social media platforms. The findings of this analysis have the potential to offer valuable perspectives for content creators, marketers, and social media strategists in navigating the complex and ever-evolving landscape of Reddit.

1 Introduction

In the ever-evolving landscape of data analytics, our project embarks on a comprehensive exploration of the Reddit Submissions dataset, a rich repository that captures the intricacies of content sharing and engagement within the expansive Reddit community. Spanning from July 2008 to January 2013, this dataset offers a unique lens into the dynamics of reddit submissions, shedding light on the evolution of content over time.

With a staggering size of almost 14 GB, the Reddit Submissions dataset presents a formidable challenge in terms of both storage and processing. To tackle this, we have harnessed the capabilities of cutting-edge cloud services, specifically focusing on the robust features of Amazon Web Services (AWS). Our arsenal includes powerful tools such as Apache Spark, AWS Elastic Kubernetes Service (EKS), AWS Simple Storage Service (S3), and AWS Elastic MapReduce (EMR), among others.

The heart of our analytical framework lies in the distributed systems architecture, exemplified by Apache Spark, which enables parallel processing and scalability. This approach not only ensures efficient handling of the colossal dataset but also unlocks the potential for complex analyses and insights extraction.

2 System and Services

2.1 AWS S3

Amazon Simple Storage Service (Amazon S3) is an object storage service offering industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps. With cost-effective storage classes and easy-to-use management features, you can optimize costs, organize data, and configure fine-tuned access controls to meet specific business, organizational, and compliance requirements.

It's designed to be simple, scalable, and secure, making it an ideal solution for storing and retrieving any amount of data from anywhere on the web. With Amazon S3, you can easily store files like documents, photos, videos, and backups. It's like having a huge, secure hard drive available on the internet.

You can use it to keep your data safe, share files with others, and even host static websites. S3 is known for its durability, high availability, and flexible pricing, making it a popular choice for businesses and individuals looking for reliable cloud storage

Figure 1: AWS S3

2.2 AWS EKS

Amazon Elastic Kubernetes Service (Amazon EKS) is a managed Kubernetes service to run Kubernetes in the AWS cloud and on-premises data centers. In the cloud, Amazon EKS automatically manages the availability and scalability of the Kubernetes control plane nodes responsible for scheduling containers, managing application availability, storing cluster data, and other key tasks. With Amazon EKS, you can take advantage of all the performance, scale, reliability, and availability of AWS infrastructure, as well as integrations with AWS networking and security services. On-premises, EKS provides a consistent, fully supported Kubernetes solution with integrated tooling and simple deployment to AWS Outposts, virtual machines, or bare metal servers.

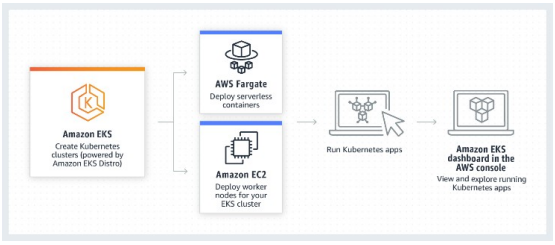

Figure 2: AWS EKS

2.3 AWS EMR

Amazon EMR is the industry-leading cloud big data solution for petabyte-scale data processing, interactive analytics, and machine learning using open-source frameworks such as Apache Spark, Apache Hive, and Presto. EMR on EKS is a deployment option in EMR that allows to automate the provisioning and management of open-source big data frameworks on EKS. There are several advantages of running optimized spark runtime provided by EMR on EKS such as 3x faster performance, fully managed lifecycle of these jobs, built-in monitoring and logging functionality, integrates securely with Kubernetes and many more. Because Kubernetes can natively run Spark jobs, if you use multi-tenant EKS environment, your spark jobs are deployed in seconds vs minutes when compared to EC2 based deployments

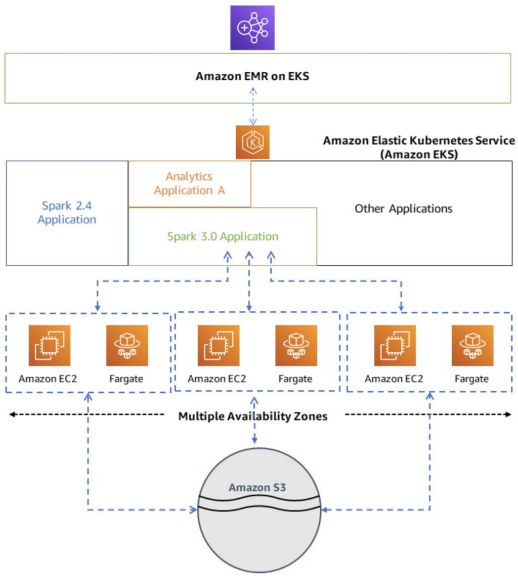

Figure 3: EMR on EKS works with other AWS services

2.4 Apache Spark

Apache Spark on AWS EKS (Elastic Kubernetes Service) combines the powerful data processing capabilities of Spark with the scalability and flexibility of AWS's managed Kubernetes service. In simple terms, Apache Spark is a popular tool used for handling large-scale data processing and analytics, offering speed and ease of use. AWS EKS, on the other hand, is a cloud service provided by Amazon that allows you to run Kubernetes, a system for automating the deployment, scaling, and management of applications, without the complexity of managing the infrastructure yourself. By integrating Spark with AWS EKS, users can easily deploy and manage Spark applications in a highly scalable and flexible cloud environment. This setup is ideal for businesses and developers who need to process large volumes of data quickly and efficiently, without worrying about the underlying infrastructure. The combination of Spark's data processing power with the robustness and scalability

of AWS EKS makes it a compelling choice for modern data-driven applications.

3 Demo Scenario

3.1 Data Collection and Preprocessing:

Collect a dataset from a social media platform. Use AWS EMR to preprocess the data. This involves cleaning, normalizing, and structuring the data to be suitable for analysis.

3.2 Setting up the Environment:

Activate the AWS Elastic Kubernetes Service (EKS) cluster, which will host the Spark application. Configure the environment to ensure seamless integration between AWS services and the Spark application.

3.3 Running the Sentiment Analysis:

Deploy a Python script on AWS EKS that utilizes Apache Spark for distributed processing. The script analyzes the sentiment of the social media posts, categorizing them as positive, negative, or neutral.

3.4 Monitoring and Logging

Utilize AWS CloudWatch to monitor the performance and health of the EKS cluster and Spark application. Access logs and metrics through the AWS console to observe the distributed system's features in action, such as load balancing, auto-scaling, and fault tolerance.

4 Distributed Characteristics

4.1 Auto-Scaling (EKS):

EKS (Elastic Kubernetes Service) supports Kubernetes auto-scaling, a feature that dynamically adjusts the number of nodes within your clusters based on the current workload demands. This capability ensures efficient resource utilization, optimizing costs while providing the scalability needed to handle varying levels of application traffic and workload. In practical terms, this means that during peak times, EKS can automatically scale up to meet increased demand, and scale down during off-peak hours to minimize costs.

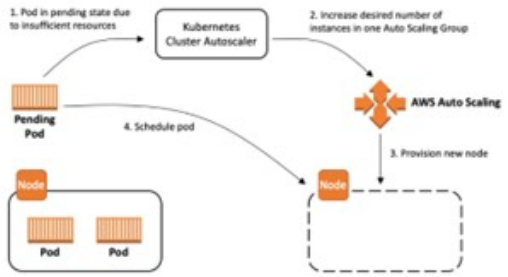

Figure 4: EKS Cluster Auto-Scaler

4.2 High Availabilty (EKS):

It is a part of Fault Tolerance. High availability in EKS is achieved by spanning Kubernetes clusters across multiple AWS Availability Zones. This design ensures that if one zone experiences issues, such as hardware failures or network disruptions, the other zones continue to function seamlessly. This redundancy minimizes potential service downtime and maintains application availability, which is critical for business continuity and customer trust.

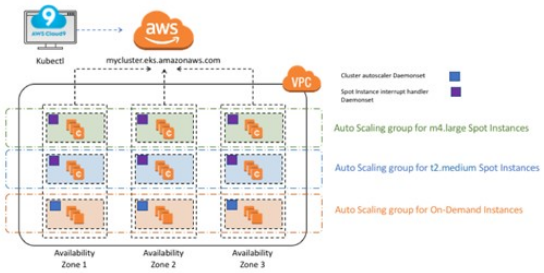

Figure 5: High Availabilty in EKS

4.3 Elasticity (EMR Systems):

AWS EMR (Elastic MapReduce) systems exhibit elasticity by dynamically scaling computing resources up or down based on demand. This feature is especially crucial in healthcare settings, where workloads can vary significantly. For instance, during health crises or data-intensive research periods, EMR systems can automatically increase resource allocation to handle the increased load, ensuring consistent performance.

4.4 Data Redundancy (EMR Systems

It is a part of Fault Tolerance and Replication. AWS provides robust data redundancy for its EMR systems. This redundancy is vital in preventing data

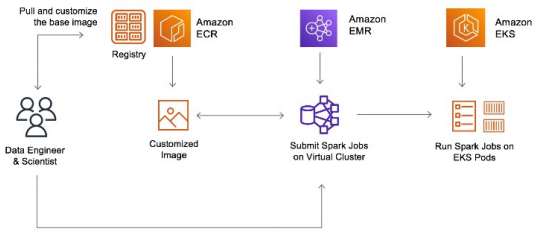

Figure 6: Elasticity in EMR System

loss and ensuring the continuous operation of systems, contributing significantly to their fault tolerance. In healthcare, where data integrity and availability are paramount, this redundancy means that even in the event of a hardware failure, data remains secure and accessible

4.5 Security and Compliance (EMR Systems):

EMR systems on AWS are designed with a strong emphasis on security and compliance, particularly for sectors like healthcare, which require adherence to regulations such as HIPAA (Health Insurance Portability and Accountability Act). This involves data encryption, secure network configurations, and compliance with healthcare standards. AWS EKS has also improved its control plane scaling and update speed, which indirectly enhances the security and compliance aspect by ensuring the system is up-to-date with the latest security patches and features.

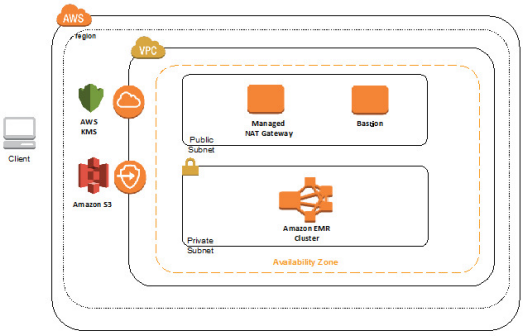

Figure 7: Security and Compliance

5 Results

Upon executing the kubectl command in our EMR on EKS setup, the system initiated the sentiment analysis process and efficiently handled the task of categorizing Reddit submissions into positive, negative, and neutral sentiments. The results of this analysis were systematically saved into a CSV file, offering a clear, structured view of the sentiments across various posts.

Concurrently, detailed logs of the process were generated and made accessible through AWS CloudWatch. These logs provided real-time insights into the orchestration of containers and pods within the Kubernetes environment, demonstrating the seamless creation and management of these resources.

14 points	funny	illuminatedwax	2008-11-16T18:4	positive
22 points	reddit.com	spez	2009-02-11T05:5	negative
76 points	WTF	masta	2008-12-02T07:1	neutral
7 points	funny	illuminatedwax	2008-11-16T18:4	neutral
11 points	funny	illuminatedwax	2008-11-16T18:4	neutral
3 points	funny	illuminatedwax	2009-01-10T08:1	neutral

Figure 8: Result from our project

6 Conclusion

In conclusion, this project successfully leveraged advanced cloud technologies such as Apache Spark, AWS S3, AWS EKS, and AWS EMR to conduct a detailed sentiment analysis on Reddit data. Utilizing the scalability and efficiency of AWS services, we efficiently processed large datasets, uncovering key insights about user sentiments influenced by factors like timing, subreddit themes, and engagement levels. These findings offer valuable guidance for content strategy and engagement on social media, showcasing the significant role of cloud computing and big data analytics in understanding and navigating the complexities of online platforms.

References

[1] Amazon Web Services, Inc. "Amazon EMR on EKS," *Amazon EMR on EKS Development Guide*, [Online]. Available: https://docs.aws.amazon.com/emr/latest/EMR-on-EKS-DevelopmentGuide/emr-eks-overview.html.

[2] V. Sharma, "EMR Cluster Setup on EKS," *Medium*, [Online]. Available: https://medium.com/@vinodsharmamui/emr-cluster-setup-on-eks-a11dd0b6cb2a, 2023.