Comparative Analysis of Cloud Platforms: Evaluating Performance and Scalability for Machine Learning Models

Dhairyashil Shendage[1], Kartik Jagtap[2], Saksham Sharma[3], Pranjal Shinde[4]
[1](Affiliation): 21bds017 , Indian Institute of Information Technology Dharwad
[2](Affiliation): 21bds025 , Indian Institute of Information Technology Dharwad
[3](Affiliation): 21bds058 , Indian Institute of Information Technology Dharwad
[4](Affiliation): 21bds062 , Indian Institute of Information Technology Dharwad
Under the Guidance of : Dr. Animesh Chaturvedi

I. ABSTRACT

Achieving optimal performance and scalability in Platform as a Service (PaaS) and Software as a Service (SaaS) Cloud environments is crucial for various computational tasks. In this study, we focus on assessing the performance and scalability of different cloud platforms through the deployment and training of an LLM (Large Language Model) model. Specifically, we conduct experiments on Amazon AWS, GPU instances, and Kaggle's cloud infrastructure. Through rigorous qualitative and quantitative analysis, we evaluate various aspects including performance metrics, scalability, deployment features, tuning time, and other relevant factors. The objective is to provide insights that facilitate the selection of the most suitable cloud platform for LLM tasks. By comparing [3] and contrasting the performance of these platforms, this research aims to offer valuable guidance to practitioners and researchers in making informed decisions regarding cloud platform selection for their LLM workloads.

II. INTRODUCTION

Cloud computing has revolutionized the way businesses and individuals access and utilize computing resources. Platform as a Service (PaaS) and Software as a Service (SaaS) are two prominent models in cloud computing [1] that offer scalable and flexible solutions for diverse computational tasks. In this study, we delve into the assessment of these cloud paradigms, focusing on their performance and scalability for the deployment and training of a sentiment analysis model.

Sentiment analysis, a branch of Natural Language Processing (NLP), plays a pivotal role in understanding the sentiment expressed in textual data [4]. In our scenario, we employ an LLM (Large Language Model) for sentiment analysis, specifically targeting movie reviews. The objective is to train and test the model to accurately determine the sentiment of the reviews as positive, negative, mixed, or neutral.

Our evaluation encompasses several prominent cloud platforms, including Amazon AWS, Kaggle, Google cloud(GCP) and Google Colab. Specifically, we leverage the comprehensive suite of services offered by Amazon AWS, integrating Amazon SageMaker for model training, Amazon Simple Storage Service (S3) for data storage, and Amazon Comprehend for natural language processing tasks. Additionally, we utilize Kaggle's cloud infrastructure and Google Colab's T4 GPU and CPU resources for comparative analysis.

In this study, we analyze cloud platforms, evaluating scalability, feature accessibility, deployment options, and billing methods. Our goal is to offer valuable insights to practitioners and researchers, facilitating informed decisions in selecting cloud platforms for sentiment analysis tasks.

III. CLOUD SERVICES

A. General Cloud Architecture and Platforms

*1) **Amazon Web Services**:* Among the array of services in AWS, we relied on S3 for scalable object storage, SageMaker for machine learning model training and deployment, IAM for managing access to AWS resources, and Comprehend for natural language processing tasks such as sentiment analysis

AWS Architecture encompasses a range of services across computing storage, database and networking that work together to form a resilient and efficient cloud environment. The architecture of AWS is designed to provide a mix of Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) offerings.

- SageMaker [5]: Managed ML service for model development and deployment.
- S3: Versatile object storage with encryption.
- Comprehend: NLP service for sentiment analysis and entity recognition.
- IAM: Access control and encryption management.
- EC2: Scalable virtual servers for applications.

*2) **Google colab**:* Google Colab, short for "Colaboratory," is a cloud-based service that allows users to write and execute Python code through their browser and provides free access to computing resources, including GPUs and TPUs. Colab is

especially well suited to machine learning, data science, and education.

*3) **Google cloud Platform**:* An array of cloud-based tools and services are available from Google Cloud Platform (GCP), such as Google Colab for creating and sharing machine learning models. GCP offers a collaborative environment for developing and implementing complex models, such as Large Language Models (LLMs) like BERT for natural language processing jobs, with features like Colab Notebooks. Platform as a Service (PaaS) is how GCP is categorized; it offers users a complete platform for creating, managing, and executing cloud applications with ease.

B. Custom Cloud Platform Usage

*1) **Amazon AWS Cloud**:* In Amazon AWS, custom procedures were implemented using the Amazon SageMaker Notebook Instance. Initially, a Python script was developed within the Jupyter Lab environment for reading and preprocessing the data from the "Walmart Data.csv" file. Subsequently, integration with Amazon Simple Storage Service (S3) was established to facilitate efficient data storage and access. The integration process involved connecting the SageMaker notebook instance with the S3 storage buckets. This connection was achieved using the Boto3 library, as illustrated below:

Uploading files to S3 buckets and Furthermore, the Amazon Comprehend API was utilized for sentiment detection tasks.

i. Sentiment Detection Job using Amazon Comprehend:

We conducted sentiment analysis using Amazon Comprehend's API, which retrieves data from Amazon S3 and executes sentiment detection jobs. Initially labeled with positive and negative tags, the data undergoes training. Following this process, we receive performance metrics, memory and CPU utilization data, and a CSV output file with predicted sentiment labels (positive, negative, mixed, neutral). This facilitates professional real-time classification tasks.

SageMaker Training feature integrated in the Sentiment Detection Job is used, providing our execution model's ARN number,dataset location at prefix, and we obtain the Utilization,Performance Metrics from Amazon CloudWatch.

Pricing and Cost	
Pay-as-you-go	
Resource Utilization	
Disk Utilization	0.55%
CPU Utilization	55%
GPU Utilization	1.2%
RunTime	
RunTime	8 minutes, 22 seconds
Performance Metrics	
Accuracy	0.59
Precision	0.8352954910844783
Recall	0.59
F1 Score	0.6890849686162005

TABLE I: Quantitative analysis for Amazon Comprehend service

ii. Custom Classification (Endpoint usage):

The Custom Classification Job serves the purpose of categorizing documents, such as our dataset in CSV format, into specific classes or labels. This job utilizes the Naive Bayes Classifier for training the data. Additionally, it involves the creation of endpoints, enabling us to test the model's performance on any test data by inputting text and receiving the predicted sentiment. We executed this job and deployed it on the AWS Endpoint for real-time analysis, which includes reporting on loss percentages and providing metrics on utilization and performance.

These are the results obtained after training on the custom classification job of amazon comprehend.

Performance metrics	
Accuracy	90%
Recall	90%
F1 Score	90%
Precision	91%

TABLE II: Performance Metrics of Custom Classification Job

iii. Deployment on Amazon EC2:

We begun by starting an Amazon EC2 instance and choosing the Ubuntu Server 22.04 (64-bit) AMI to use as our virtual server in order to deploy our sentiment analysis model. We chose a t2.micro instance type because it met our needs for the size of our application—1 CPU and 1GB of RAM. We created a new Key Pair in.pem format with RSA encryption to guarantee safe access to the instance. This Key Pair acts as our secure login mechanism for the EC2 instance.

We then set up a security group to control both incoming and outgoing traffic to our EC2 instance [2]. We painstakingly defined rules to grant and deny access based on predetermined IP addresses. After setting up security settings, we used WinSCP to make it easier to move important files—like our sentiment analysis model files—from our local computer to the EC2 instance. By taking this step, you could be sure that the server had all the resources needed for deployment and execution.

In addition, we used Putty to create an SSH connection to the EC2 instance, which gave us safe terminal access for further operations. We successfully deployed our model in a real-time Flask web application environment by running our sentiment analysis model files on the EC2 instance's terminal after completing these preparatory steps. Throughout this process, our sentiment analysis model was securely and smoothly deployed on Amazon EC2 thanks to close attention to resource allocation, security protocols, and file transfer mechanisms.

Our localhost and the service can be accessed - here

*2) **Kaggle**:* We developed and implemented a sentiment analysis model in a notebook setting using Kaggle as a Platform as a Service (PaaS). We trained our sentiment analysis model with a sentiment analyzer by uploading a dataset of sentiment reviews and making use of Kaggle's cloud features. We were able to effectively utilize the notebook's computational resources by running it on Kaggle. Furthermore, we were able to extract performance and utilization metrics from our model

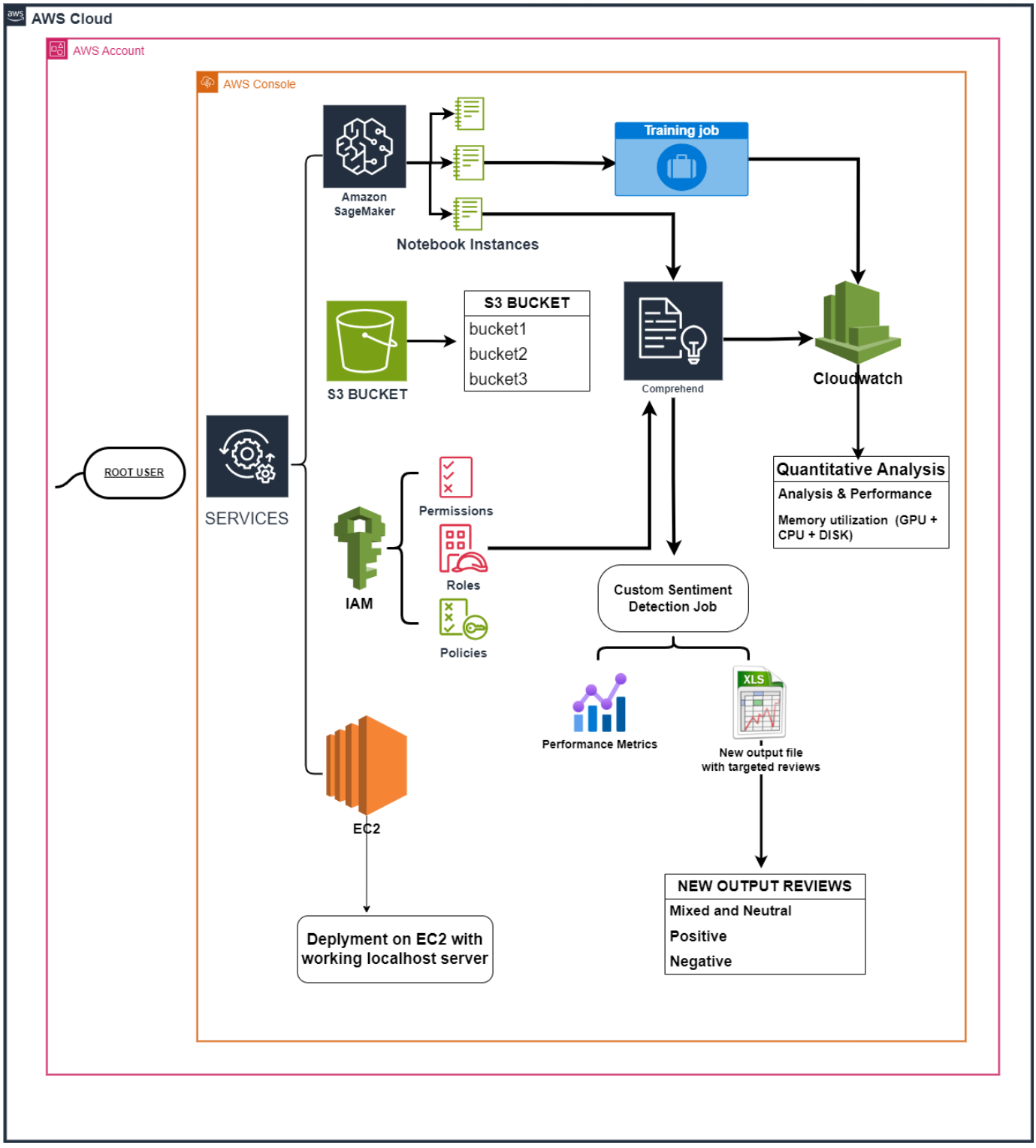

Fig. 1: AWS flow diagram

execution with ease, which allowed us to perform thorough comparative analyses.

Adding Kaggle Cloud functionality greatly expedited our process. We profited from a cohesive environment for data exploration, model development, and analysis. Overall, Kaggle Cloud helped us become more productive and allowed us to thoroughly assess our sentiment analysis model. This allowed us to make decisions based on detailed performance and utilization metrics.

The following is the Quantitative Analysis Table for the Kaggle:

Pricing and Cost	
Free	
Resource Utilization	
CPU usage	0.5%
Memory Utilization:	
Total	31.36 GB
Available	30.25 GB
Used	0.66 GB
Free	24.12 GB
RunTime	
RunTime	62.85114336013794 seconds
Performance Metrics	
Accuracy	0.6678339169584793
Precision	0.6263157894736842
Recall	0.833
F1 Score	0.7150214592274677

TABLE III: Quantitative analysis for Kaggle

*3) **Google Colab GPU**:* Using the powerful T4 GPU in Google Colab, we started training right in the notebook, allowing for easy integration with Weights & Biases (WandB) for in-the-moment tracking of important metrics like accuracy and loss. We dynamically managed computing resources, ensuring flexibility to adjust to changing workloads, by utilizing the cloud-native characteristics of the T4 GPU. This simplified method not only increased productivity but also made it easier for team members to work together seamlessly, which promoted real-time insight sharing and group model optimization. All things considered, our use of the T4 GPU in Google Colab in conjunction with WandB's monitoring capabilities created a simple yet effective training environment for machine learning projects.

Pricing and Cost	
Free	
Resource Utilization	
CPU usage	54.2 %
Memory Utilization:	
Total	12.67 GB
Available	11.24 GB
Used	1.15 GB
Free	7.17 GB
RunTime	
RunTime	20.439889430999756 seconds
Performance Metrics	
Accuracy	0.6683341670835418
Precision	0.6267870579382995
Recall	0.833
F1 Score	0.7153284671532847

TABLE IV: Quantitative analysis for Google GPU

*4) **Google Cloud Platform**:* Google Cloud Platform (GCP) is a comprehensive suite of cloud computing services offered by Google, providing infrastructure, platform, and software services for building, deploying, and managing applications and data [6]. With scalable and reliable computing resources, advanced analytics, and AI capabilities.

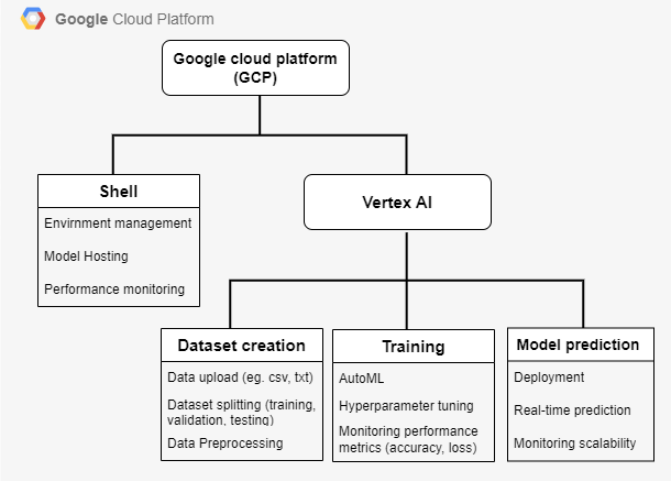

Fig. 2: Google cloud workflow

i. Cloud Shell Terminal and Performance Metrics:

In our Google Cloud Platform (GCP) project, we utilize the Cloud Shell for managing our environment, accessing resources, and executing commands, offering a convenient and efficient interface without complex setup. Additionally, we host and run our sentiment analysis model within the Cloud Shell, ensuring seamless integration with GCP's infrastructure. We also gather performance metrics, including latency and resource utilization, guiding optimization for optimal model performance.

ii. Vertex AI - AutoML service with batch prediction

In our sentiment analysis project on Google Cloud Platform, we maximize efficiency by integrating Vertex AI's AutoML capabilities into our workflow. AutoML simplifies and automates key aspects of model development, starting with model creation and training using preprocessed datasets. It autonomously selects the optimal model architecture, saving us time and effort in manual exploration. Additionally, AutoML automates hyperparameter tuning, enhancing our model's performance by iteratively adjusting parameters based on feedback. Once training is complete, AutoML facilitates seamless deployment of the trained model using Vertex AI's infrastructure. This deployment process establishes real-time prediction endpoints, ensuring our sentiment analysis model is readily accessible for use in production environments with minimal setup. By leveraging both the hosting and monitoring capabilities of the Cloud Shell and the automation provided by AutoML within Vertex AI, we build a robust and efficient sentiment analysis solution on Google Cloud Platform.

Attribute	Local Device	Google Colab (T4 GPU)	Kaggle	Amazon Web Services (AWS)
Pricing	Free	Free	Free	Pay-as-you-go
Compute Resources				
CPU Specs	Intel i5-11400h	Intel Xeon	Intel Xeon	EC2 Instance Types
CPU Cores	6	2	2	Varies
CPU Clock Speed (GHz)	2.7	2.2	2.2	Varies (2.3 - 3.5 GHz)
GPU Specs	GTX-1650	NVIDIA Tesla T4	N/A	EC2 GPU Instances
GPU Memory Size (GB)	N/A	16	N/A	Varies (8 - 32 GB)
Storage Options				
Free Storage (GB)	15	15	Varies	Varies (5 - 50 GB)
Networking				
Bandwidth (Mbps)	High	High	High	High (100 - 1000 Mbps)
Latency (ms)	Low	Low	Low	Low (5 - 50 ms)
Scalability				
Vertical Scaling	Limited	Limited	Limited	Flexible
Horizontal Scaling	Limited	Limited	Limited	Flexible
Security				
Data Encryption	SSL/TLS	SSL/TLS	SSL/TLS	AES Encryption
Access Control	Google Account	Google Account	Kaggle Account	IAM (Identity and Access Management)
Support and Documentation				
Technical Support (%)	60	60	60	Documentation, Premium Support
Developer Tools (%)	70	70	70	SDKs, APIs, Developer Tools
Ecosystem and Integration				
Compatibility (%)	90	90	90	Various Languages, Frameworks
Marketplace (%)	N/A	N/A	N/A	AWS Marketplace
User Experience				
Ease of Use (%)	80	80	80	Management Console, UI
Learning Curve (%)	70	70	70	Moderate

TABLE V: Comparison of Cloud Platforms

*5) **Local CPU**:* For our project, using the local CPU not only provided a more affordable option, but it also guaranteed that resources would be available when needed. The ensuing quantitative study explores the memory usage and performance metrics obtained with this configuration, providing insight into the effectiveness of using local resources for our project's tasks. This thorough evaluation seeks to highlight the importance of locally accessible resources in promoting project success by offering subtle insights into the efficacy and efficiency of our selected strategy.

Pricing and Cost	
Free	
Resource Utilization	
CPU usage	1.4 %
Memory Utilization:	
Total	7.78 GB
Available	1.16 GB
Used	6.62 GB
Free	1.16 GB
RunTime	
RunTime	9.51947021484375 seconds
Performance Metrics	
Accuracy	0.6683341670835418
Precision	0.6267870579382995
Recall	0.833
F1 Score	0.7153284671532847

TABLE VI: Quantitative analysis for Local CPU

IV. Conclusion and Discussions

After evaluating various platforms, it's evident that each offers unique benefits depending on project requirements. For small-scale projects with limited resources and no added costs, the Local CPU is a practical choice. Colab T4 GPU is advantageous for GPU-accelerated machine learning tasks without expenses. Kaggle provides a supportive environment for tasks with moderate resource needs and competitions, offering free GPU access. Amazon AWS Comprehend stands out for text analysis tasks with high resource availability on a pay-as-you-go model. However, for intensive machine learning training, For intensive machine learning training, Amazon EC2 instances are preferred, However, it's important to keep a close eye on costs when using EC2 instances for intensive machine learning training. In conclusion, selecting the appropriate platform should align with project needs and budget constraints to ensure efficiency and cost-effectiveness.

References

[1] Mohammad Ubaidullah Bokhari, Qahtan Makki Shallal, and Yahya Kord Tamandani. Cloud computing service models: A comparative study. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pages 890–895, 2016.

[2] Anurag Choudhary, Pradeep Kumar Verma, and Piyush Rai. The proposed pre-configured deployment model for amazon ec2 cloud services. In *2022 6th International Conference on Electronics, Communication and Aerospace Technology*, pages 794–799, 2022.

[3] Noman Islam and Aqeel-ur Rehman. A comparative study of major service providers for cloud computing. 09 2013.

[4] Anupriya Koneru, Nerella Bala Naga Sai Rajani Bhavani, K Purushottama Rao, Garikipati Sai Prakash, Immadisetty Pavan Kumar, and Velimala Venkat Kumar. Sentiment analysis on top five cloud service providers in the market. In *2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, pages 293–297. IEEE, 2018.

[5] David Nigenda, Zohar Karnin, Muhammad Bilal Zafar, Raghu Ramesha, Alan Tan, Michele Donini, and Krishnaram Kenthapadi. Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3671–3681, 2022.

[6] Pramod Singh. Deploy machine learning models to production. *Cham, Switzerland: Springer*, 2021.