Analyzing the Usability, Performance, and Cost-Efficiency of Deploying ML Models on BigQuery ML and Vertex AI in Google Cloud

Hongyu Wang
Department of Computational, Engineering, & Mathematical Sciences, Texas A&M University-San Antonio
USA
hwang@tamusa.edu

Jeong Yang
Department of Computational, Engineering, & Mathematical Sciences, Texas A&M University-San Antonio
USA
jyang@tamusa.edu

Gongbo Liang
Department of Computational, Engineering, & Mathematical Sciences, Texas A&M University-San Antonio
USA
gliang@tamusa.edu

Young Lee
Department of Computational, Engineering, & Mathematical Sciences, Texas A&M University-San Antonio
USA
ylee@tamusa.edu

Zechun Cao
Department of Computational, Engineering, & Mathematical Sciences, Texas A&M University-San Antonio
USA
zcao@tamusa.edu

Abstract

This study compared and analyzed the usability, performance, and cost-efficiency of deploying Machine Learning (ML) models in two ML-AI platforms in Google Cloud: BigQuery ML and Vertex AI. Through the experiments with two separate cases, the analysis was conducted with MIMIC-IV datasets of hospitalized patients to deploy regression models on each platform to predict mortality and progression of diseases. The documentation, learning curve, and resource suitability of the platforms were evaluated to access their usability. The study evaluated the total running times and resource utilizations, including storage and compute, to analyze their performance and cost efficiency. The analysis results showed that BigQuery ML offers good usability with easy-to-follow documentation and a moderate learning curve for cloud users, making it more suitable for SQL-savvy users and large-scale data analytics tasks. It also showed efficient resource management and deployment despite its higher initial processing times during the training. Vertex AI incurred higher costs due to longer training times and specific resource allocations. The findings indicate that BigQuery ML seems to be more efficient, particularly in terms of processing time and cost for the experimented clinical dataset and regression models, emphasizing its suitability for large-scale data processing tasks where efficiency is essential.

CCS Concepts

• **Networks**; • **Network services**; • **Cloud computing**;

ICCBDC 2024, August 15–17, 2024, Oxford, United Kingdom

ACM ISBN 979-8-4007-1725-3/24/08
https://doi.org/10.1145/3694860.3694863

Keywords

Cloud Computing, Machine Learning, AI, Artificial Intelligence, Deployment Architectures, Google Cloud Platform, GCP, Performance Analysis, Usability Evaluation, Cost Efficiency, BigQuery ML, Vertex AI, MIMIC-IV

ACM Reference Format:
Hongyu Wang, Jeong Yang, Gongbo Liang, Young Lee, and Zechun Cao. 2024. Analyzing the Usability, Performance, and Cost-Efficiency of Deploying ML Models on BigQuery ML and Vertex AI in Google Cloud. In *2024 8th International Conference on Cloud and Big Data Computing (ICCBDC 2024), August 15–17, 2024, Oxford, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3694860.3694863

1 Introduction

Cloud computing has transformed the operational capabilities of various sectors by offering scalable, flexible platforms that enable efficient data processing and storage solutions. Machine Learning (ML) and Cloud Computing have become popular and widely used in the tech world. They both can work together to bring several benefits to organizations as the cloud offers many ML solutions that help manage AI applications [1]. The convergence of cloud computing with ML technologies has catalyzed significant innovations, empowering businesses, and researchers to harness sophisticated data analysis and predictive modelling techniques that were once beyond reach. This integration is particularly impactful in areas in healthcare, finance, and manufacturing, where the ability to rapidly process and analyze large datasets can result in substantial improvements in efficiency and effectiveness [2, 3]. The on-demand nature of utilizing cloud resources is also known to be faster and more efficient than operating hardware and deploying and updating localized software [4–6] with the availability of computer system resources, particularly data storage and computing power, without direct active management by the user. Artificial intelligence (AI) is one of the most transformative technologies that has been revolutionizing industries and enterprises around the world. When AI is combined with Cloud Computing, the way businesses operate, store, and process data undergoes a remarkable revolution. AI in cloud computing has provided various advantages to businesses and industries, including better efficiency, cost savings, scalability, and productivity [7]. AI has become more affordable with the use of cloud-based AI-ML services such as cheap data storage, AI-enabled SDKs, and built-in APIs [5, 8]. With these, organizations are moving to hybrid or fully cloud to follow the evolving requirements of their business needs [3, 9], utilizing ML and AI tools and applications.

Major cloud computing services include Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) [10]. These models allow organizations to avoid upfront infrastructure costs, focus on core business activities, and scale resources as needed [11]. Cloud service providers such as AWS, Microsoft Azure and Google Cloud offer a range of ML algorithms and AI technologies to enable users to build and deploy intelligent applications in the cloud. Such resources can be helpful for businesses and individuals who want to leverage these technologies without having to invest in their own infrastructure.

Google Cloud Platform (GCP) exemplifies these benefits by offering ML-AI options, including BigQuery ML for both data warehousing and data analytics and Vertex AI for advanced machine learning. BigQuery ML is a fully managed data warehouse that allows running fast SQL queries using the processing power of Google's infrastructure. It is designed to handle large datasets and enables users to execute complex queries quickly. It provides two services in one: both storage and analytics. BigQuery ML extends these capabilities by allowing users to create and execute machine learning models using SQL queries directly within the data warehouse environment, facilitating seamless integration between data storage and analysis [12].

Vertex AI, on the other hand, is a comprehensive machine learning platform that provides tools for training and deploying machine learning models. It supports AutoML, which automates the process of building high-quality models, and also allows for custom model training, offering flexibility for various machine learning tasks. Vertex AI integrates with other GCP services, making it a robust choice for scalable and end-to-end machine learning workflows [13]. Recent technological advancements in cloud-based machine learning platforms, such as Google Cloud Platform's BigQuery ML and Vertex AI, demonstrate a growing trend towards more integrated, user-friendly solutions. These tools not only enhance the capabilities of data scientists but also broaden the accessibility of ML technologies to non-experts, promoting wider adoption across industries [14]. The evolution of these platforms reflects a significant shift towards democratizing advanced analytics, making it a pivotal element in competitive business strategies and informed decision-making processes.

As the complexity of machine learning applications increases, the selection of an appropriate deployment architecture becomes very important. Each architecture offers unique advantages and challenges, especially concerning performance metrics such as speed and efficiency as well as cost-efficiency on total expenses, covering compute time, and storage. The challenges come here in navigating these architectural choices to optimize for specific needs and strategic goals for individuals and businesses. The decision-making process is critical, as the selected deployment strategy significantly impacts the project's success and scalability toward minimal cost [14].

This study aims to compare and evaluate the efficiency of two ML-AL platforms on deploying ML models on GCP: BigQuery ML and Vertex AI. Specifically, the objectives include:

- Evaluating the usability of each deployment strategy, focusing on ease of documentation, management simplicity, and the learning curve for new users.
- Assessing the performance of integrated database functions versus dedicated machine learning services in terms of processing time, storage usage, and training time.
- Evaluating the cost efficiency of deployment strategies, focusing on overall expenses with compute time, and storage.

The contributions of this research are threefold:

- Presenting systematic analysis through empirical experiments on the performance and cost-efficiency of popular ML deployment architectures on GCP.
- Developing a set of best practices for deploying ML models in cloud environments to enhance operational efficiency and user experience.
- Providing guidelines that help individuals and organizations choose the most suitable deployment architecture, optimizing both the cost-effectiveness and functional effectiveness of their cloud-based ML initiatives. [15, 16]

2 Related Work

This section provides a literature review of previous work on cloud-based machine learning deployment strategies, their performance metrics, and cost-efficiency, and identifies gaps that this research aims to fill. Several studies have highlighted the transformative impact of cloud computing on machine learning. These studies emphasize the enhanced computational power, storage capabilities, and scalability that cloud platforms offer, enabling more complex and data-intensive machine learning models to be deployed efficiently [17, 18]. Hashem et al. [3] discuss the rise of big data analytics in cloud environments and outline the opportunities and challenges associated with integrating machine learning technologies. Google Cloud Platform (GCP) provides a robust infrastructure for developing and deploying AI solutions. GCP offers a comprehensive suite of tools and services that facilitate various stages of AI workflows, from data preparation and model training to deployment and monitoring. GCP's key services that support AI include BigQuery ML, a scalable data warehouse with integrated machine learning and analytic capabilities, and Vertex AI, a unified platform for building, deploying, and scaling ML models [12, 13]. These platforms are designed to streamline AI workflows and make advanced analytics accessible to a wider audience. Studies have shown that the integration of GCP with AI workflows enhances productivity, scalability, and the ability to handle large-scale data processing tasks [19, 20].

Deployment strategies for machine learning models on cloud have been a critical area of research. Polyzotis et al. [14] specifically address the data management challenges when deploying machine learning models at scale, providing insights into the complexities of managing data and model lifecycle in production environments. These insights are crucial for understanding the performance implications of different deployment architectures. Additionally, the utilization of advanced techniques such as the analytical hierarchy process helps in evaluating various cloud platforms to determine the most suitable one for specific data science workflows, emphasizing a structured decision-making approach. Performance metrics and cost-efficiency are paramount in evaluating the effectiveness of machine learning deployment strategies. Studies often focus on specific metrics such as processing speed and resource utilization [21, 22]. Cost-efficiency involves analyzing the total expenses associated with compute time, and storage. Understanding both performance and cost-efficiency helps practitioners optimize their deployment strategies for both operational efficiency and budget constraints [23]. While the adoption of cloud-based machine learning solutions increases, there are also security concerns. Systematic reviews of security in cloud machine learning platforms highlight the prevalence of attacks such as data poisoning and model theft and emphasize the importance of robust security measures and countermeasures to protect sensitive ML/DL models [24]. The studies that integrate security measures into the deployment strategies are also needed, to ensure the integrity and confidentiality of the deployed models. While extensive research has been conducted on the technical aspects of deploying machine learning models, there are not many studies that comprehensively compare the performance and cost-efficiency of deploying ML/AI services in cloud. This gap in literature presents an opportunity for this study to contribute valuable insights into the decision-making processes involved in selecting optimal deployment strategies for cloud-based machine learning [25].

3 Methodology

The study is designed to compare and evaluate the easy-to-use, performance, and cost-efficiency of integrated database-machine learning functions on BigQuery ML versus dedicated machine learning services on Vertex AI in Google Cloud. BigQuery ML uses integrated SQL-based functions on queries to create and execute ML models while Vertex AI builds models with relatively less or no code with dedicated Auto ML capabilities and eservices. The selection is based on its comprehensive suite of machine learning and big data tools. These services are well-integrated within the GCP ecosystem, providing a robust environment for deploying, executing, and analyzing machine learning models and big data operations. This setup enables direct comparison in a controlled cloud-native environment, ensuring that the findings are relevant to current cloud-based ML deployment strategies [26]. This study used the MIMIC-IV dataset in experiments, which is a large, publicly available database of de-identified health information. The dataset includes clinical data such as demographics, lab results, diagnosis codes, and treatment records from a diverse patient population. The use of MIMIC-IV allows for a realistic simulation of clinical workflows and decision-making processes in a healthcare setting. It also provides a rich ground for testing and comparing the effectiveness of BigQuery ML and Vertex AI in handling real-world, data-intensive tasks in healthcare analytics.

Experiments were structured to assess key performance metrics such as easy-to-use, training time, and cost-efficiency like total expenses associated with storage utilization and compute times. For instance, using BigQuery ML and Vertex AI, experiments involving the MIMIC-IV dataset focused on tasks of patient mortality prediction and disease progression modeling [27]. These experiments were conducted under similar conditions. Performance monitoring and cost-efficiency evaluation were facilitated by GCP's integrated monitoring tools, which provide real-time insights into resource consumption, cost analysis, and system performance metrics [28].

The prepared datasets and Google's primary tools, and services used in this study are:

- **MIMIC-IV Datasets:** MIMIC-IV (Medical Information Mart for Intensive Care IV) dataset is a comprehensive, publicly available resource that includes de-identified health information from hospitalized patients admitted to critical care units. It provides detailed clinical data such as patient demographics, vital signs, laboratory test results, medications, and diagnostic codes, making it available for healthcare research and ML applications [29].
- **BigQuery ML:** Utilized for its capabilities in handling large datasets and executing SQL-like queries for data analysis and ML model training directly within the data warehouse environment [30, 31]
- **Vertex AI:** Employed for its advanced machine learning services that support the training and deployment of complex models using AutoML, custom ML training, and ready deployment to production environments.
- **Google Cloud Storage:** Served as the data repository for storing and retrieving large datasets efficiently, ensuring seamless integration with BigQuery ML and Vertex AI.
- **Logistic Regression:** A statistical method used for binary classification problems. It models the probability of a binary outcome based on one or more predictor variables. The logistic function, also known as the sigmoid function, is used to map predicted values to probabilities, which are then used to classify the outcome [32]. In this study, logistic regression was used within BigQuery ML to predict patient mortality and model disease progression. This method was chosen for its simplicity, interpretability, and efficiency in handling large datasets directly within the data warehouse and analytics environment.
- **Tabular Regression:** AutoML in Vertex AI is a versatile tool that supports a range of machine learning tasks, including both classification and regression. Specifically, for tabular data, Vertex AI can automatically handle different types of data transformations and model selection to optimize the machine learning pipeline. In the context of tabular regression, AutoML can be configured to solve tasks that would traditionally be handled by logistic regression models. This study used AutoML in Vertex AI in its analysis. While it may use more complex algorithms under the hood, it is known to effectively manage regression tasks, providing comparable or even superior performance to logistic regression depending on the dataset and specific requirements [33].

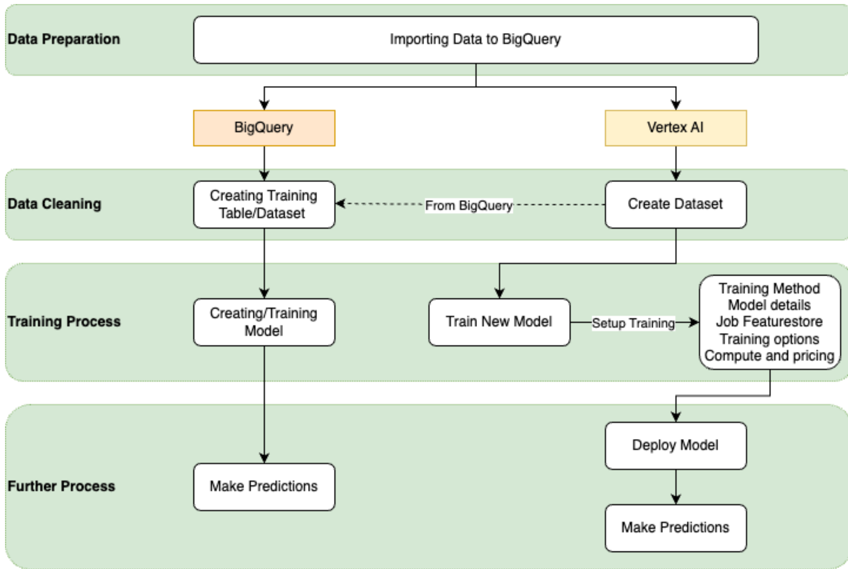

Figure 1: Flowchart Comparison of Deploying ML Models.

3.1 Data Preparation and Cleaning

To prepare the MIMIC-IV dataset for our experiments, we performed data cleaning aimed at creating two distinct tables: one for predicting patient mortality and another for modeling disease progression. Both BigQuery ML and Vertex AI used these tables to ensure a consistent comparison, analyzing their usability, performance, and cost-efficiency of deploying the same models between the two services. The first mortality table integrated patient demographic information with relevant laboratory results. This involved selecting key variables such as gender, admission type, and insurance details from the patient and admission records. We also calculated average, maximum, and minimum values for specific laboratory tests like Sodium, WBC, and Hemoglobin, which are crucial indicators of patient health. This clean and integrated dataset was used for our patient mortality prediction experiment. The table included 312,076 rows of records, with logical and physical sizes of 31.74 MB and 3.59 MB, respectively. The second disease-progression table was for modeling disease progression. It included patient demographics, lab test values over time, and outcomes such as hospital discharge status. We calculated the patient's age at admission and tracked the sequence of hospital admissions and lab test results to model the progression of diseases. This table provides a detailed temporal view of patient health, that is essential for understanding how diseases evolve over time. This table has 59,266,165 rows of records, with logical and physical sizes of 3.04 GB and 402.19 MB.

3.2 Model Deployment

BigQuery ML: The process of deploying models in BigQuery ML starts by importing the dataset into BigQuery environment, which handles large-scale data storage with querying and analytics. The entire process involves uploading the data and ensuring it is correctly formatted for analysis. In BigQuery, the data is cleaned by creating training datasets and tables. This involves organizing the data into a structured format suitable for machine learning analysis, ensuring that all necessary fields are correctly populated, and any irrelevant data is removed. Using SQL-like queries, models are created and trained directly within BigQuery. This leverages BigQuery's powerful processing capabilities to handle large datasets efficiently, optimizing the model parameters to the best performance. Once the model is trained, it is used to make predictions. These predictions can then be used for further analysis or integrated into applications to provide insights and drive decision-making . The corresponding flowchart is presented in Figure 1.

Vertex AI: Similar to BigQuery ML, the process in Vertex AI begins by importing the dataset. However, in this case, the data is prepared in BigQuery and then transferred to Vertex AI environment for further processing. In Vertex AI, a dataset is created from the imported data. This step includes defining the structure of the dataset, specifying features, and ensuring that the data is ready for training. Vertex AI offers a more flexible and powerful environment for training machine learning models. Users can select different training methods, specify model details, set up a feature store, and choose various training options. Vertex AI handles the computational aspects, optimizing the model based on the chosen settings. After training, the model can be deployed. Deployment involves setting up the model in a production environment where it can make real-time predictions. The trained model can then be used to generate predictions that are integrated into applications or systems for real-time decision-making (Figure 1).

4 Evaluation Results

The experiments on BigQuery ML and Vertex AI utilized the MIMIC-IV dataset focused on two aspects: patient mortality prediction and disease progression modeling. Each experiment tested different capabilities of the machine learning services offered by BigQuery ML

Table 1: Comparison Results on Easy-to Use.

Ease-to-Use [1]	BigQuery ML	Vertex AI
Documentation	Excellent 4.5 / 5	Excellent 4 / 5
The learning curve for a beginner	Moderate learning curve. Templates & examples available make learning easier.	Steeper learning curve. Extensive tools and features require more initial effort.
Integration with GCP services	Seamless integration with other GCP services.	Seamless integration with GCP and support for complex ML workflows.
Support for ML models	Supports SQL-based ML models, ideal for users familiar with SQL.	Supports a wide range of ML models, including AutoML and custom models.

[1] The ratings of the document and learning curve were solely based on the author's experience during the study with the two ML services in Google Cloud.

and Vertex AI. From the results of the experiments, this study analyzed the easy-to-use, performance, and cost-efficiency of deploying the same models, examining relevant metrics. The following sections present the results of the analysis on those metrics of the two services.

4.1 Results on Usability Analysis

The usability analysis focused on the easy-to-use and suitability of deploying models with two ML services, specifically from the perspective of a beginner cloud user on ML-AI tasks. This study was based on the two components regarding the usability: *System Effectiveness, which assesses the users' ability to accomplish the assigned tasks; System Efficiency, which gauges the resources required by the users to complete the tasks* [40, 41]. For this analysis on the System Effectiveness, we have evaluated the ease or difficulty of learning and performing the deployment tasks as well as the quality of documentation available on each service. In terms of System Efficiency, we have taken into consideration suitable resources for their adaptation to real-world applications. The results of this analysis are presented in Tables 1 and 2.

Through the analysis, we have discovered that BigQuery ML shows documents to be relatively easy to read/follow and offers a moderate learning curve for beginner cloud users. BigQuery ML offers a streamlined workflow for users who are familiar with SQL, allowing them to perform machine learning tasks directly within the environment. This reduces the learning effort and integration complexity, making it ideal for projects requiring fast deployment and minimal configuration. On the other hand, Vertex AI covers a broader range of machine learning tasks and user expertise, providing a versatile environment that supports both AutoML and custom model training.

While it offers more extensive tools and flexibility, it requires a higher level of user engagement and expertise to optimize model configurations and workflows, making it suitable for complex ML tasks. For the two cases in the experiments, BigQuery ML can be suitable for patient mortality prediction tasks as it involves data manipulation with required quick integration with other GCP services. Its SQL-based approach also made it particularly effective for users with SQL experience, providing a straightforward and cost-efficient solution on mortality prediction. Vertex AI can be chosen for more complex machine learning tasks, such as disease progression modeling, where its advanced capabilities and flexibility in handling custom workflows were crucial. Despite its higher cost and steeper learning curve, Vertex AI's comprehensive toolset allowed for sophisticated model training and deployment, making it ideal for scenarios demanding high accuracy and extensive customization.

4.2 Results on Performance Analysis

The performance of the services was evaluated with the training times of the same model deployments on the same datasets that were deployed on BigQuery ML and Vertex AI. On each service, training times were recorded by the monitoring services on each platform during the training. Two cases were experimented in each service:

- Case 1: patients' mortality prediction using the mortality table with 312,076 rows of records, and logical and physical sizes of 31.74 MB and 3.59 MB.
- Case 2: disease progression modeling using disease-progression table with 59,266,165 rows of records, and logical and physical sizes of 3.04 GB and 402.19 MB.

4.2.1 Performance on BigQuery ML. **Training Time:** Table 3 summarize*s* the training times (Duration) taken for two cases in BigQuery ML. Figures 2 and 3 illustrate the*ir* graphical results with the training data loss, evaluation data loss, learning rate, and duration in each iteration for both cases. For the Case 1, the total training took 11.02 seconds only throughout 1 iteration, with no learning rate because the amount of data is relatively small for the BigQuery ML. Both training data loss and evaluation data loss are reported for one iteration, showing values of 0.0256 and 0.0270 respectively. Case 2 exhibited a different pattern with the total training time, 48.53 seconds, ranging from 13.99 to 18.25 seconds across 3 iterations with the learn rate consistently increased while both training and evaluation data loss decreased. Through this analysis, we have discovered that BigQuery ML can handle large datasets more efficiently in terms of times, with a quite short processing time in Case 1 and the consistent performance in Case 2 indicates the model's ability to optimize allocating recourses and process a more complex model with larger datasets efficiently.

Model Accuracy: The results of the model deployment with linear regression are presented in Table 4. The model's accuracy was evaluated using metrics such as Mean Absolute Error (MAE), Root

Table 2: Comparison Suitability on Resources.

Feature	BigQuery ML	Vertex AI
Efficient Handling of Large Datasets	√	√
Consistent and Predictable Training Time		√
High Recall for Identifying True Positives	√	√
Precision in Reducing False Positives		√
Stable Performance Across Iterations	√	√
Flexibility in Handling Various Data Types		√
Comprehensive Automated Machine Learning Workflow		√
Ease of Integration with SQL-based Environments	√	√
Scalability for Complex Models	√	
Cost-Efficiency	√	√

Table 3: Training Times in BigQuery ML.

	Iteration	Training Data Loss	Evaluation Data Loss	Learn Rate	Duration (seconds)
Case 1	0	0.0256	0.0270	/	11.02
					Total: 11.02
Case 2	0	0.0752	0.0753	0.2	16.29
	1	0.0721	0.0728	0.4	18.25
	2	0.0711	0.0722	0.4	13.99
					Total: 48.53

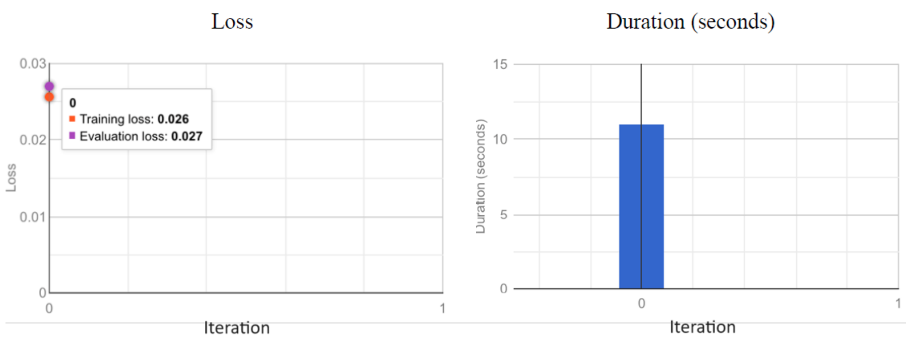

Figure 2: Graph Representation for Case 1 in BigQuery ML.

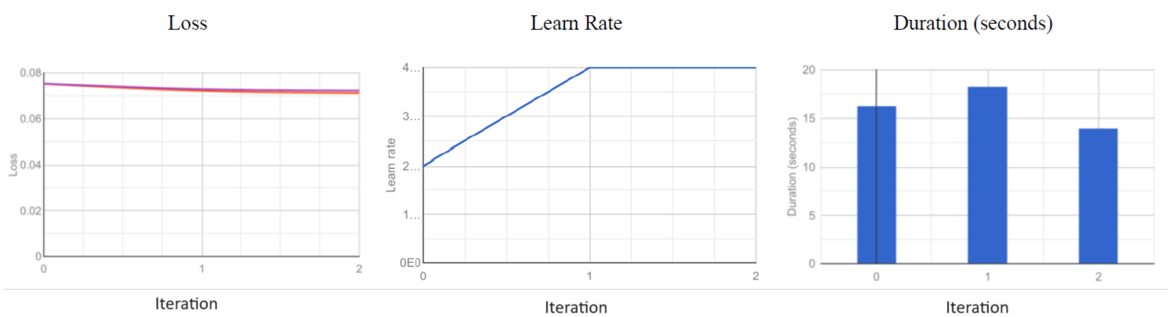

Figure 3: Graph Representation for Case 2 in BigQuery ML.

Table 4: Model Accuracy.

	Case 1		Case 2	
Metric	BigQuery ML	Vertex AI	BigQuery ML	Vertex AI
Mean Absolute Error (MAE)	0.0542	0.049	0.1532	0.106
Root Mean Squared Error (RMSE)	0.164	0.154	0.2687	0.231
R-squared	0.0285	0.093	0.0682	0.318

Table 5: Training Details in Vertex AI.

Details	Case 1	Case 2
Training time	2 hr 2 min	2 hr 9 min
Data split (Randomly assigned)	80/10/10	80/10/10
Algorithm	AutoML	AutoML
Objective	Tabular regression	Tabular regression

Mean Squared Error (RMSE), and R-squared. For Case 1, the Mean Absolute Error (MAE) is 0.0542, indicating a low average magnitude of errors. The Root Mean Squared Error (RMSE) is 0.164, showing the average magnitude of errors in a more sensitive way to larger errors. The R-squared value of 0.0285 indicates that only a small portion of the variance in the dependent variable is predictable from the independent variables, suggesting the model could be further improved. For Case 2, the Mean Absolute Error (MAE) is 0.1532, indicating a higher average magnitude of errors compared to Case 1. The Root Mean Squared Error (RMSE) is 0.2687, reflecting the errors' average magnitude in predicting the continuous outcome. The R-squared value of 0.0682 suggests that the model explains a slightly higher proportion of the variance than in Case 1 but still indicates room for improvement. Overall, the results indicate that while the model can make predictions with a certain degree of accuracy, there are significant opportunities for refining the model to improve its predictive performance, as evidenced by the relatively low R-squared values in both cases.

4.2.2 Performance on Vertex AI. **Training Time:** Vertex AI used AutoML features in its model training and deployment, focusing on regression metrics to evaluate prediction accuracy. Tabular regression was used in both experiments, aiming to predict continuous outcomes based on the prepared tabular data for the Vertex AI environment. This aligns with the clinical data from the MIMIC-IV dataset, which includes numerous numerical and categorical variables pertinent to patient outcomes. We observed that the model's results are structured differently, emphasizing error metrics which are critical for continuous data prediction. For both Cases, the training times were similar, with Case 1 taking 2 hours and 2 minutes, and Case 2 taking 2 hours and 9 minutes (Table 5) despite the size difference on their datasets, utilizing AutoML for tabular regression with an 80/10/10 data split. The consistent training times and robust data handling capabilities may implicate Vertex AI's efficiency and reliability in managing large-scale machine learning tasks. For showing the processing capabilities, Vertex AI took longer time than running those on BigQuery.

Model Accuracy: Vertex AI reporting provided the metrics of MAE (0.049) and RMSE (0.154), indicating moderate accuracy in the model's prediction for Case 1, with errors suggesting deviations from actual outcomes. The low R-squared value of 0.093 indicates that the model leaves a significant amount of variance unexplained. This shows potential areas for model refinement to better capture the complexities of the data or possibly the inclusion of more predictive features. The model displayed improved predictive accuracy with an MAE of 0.106 and RMSE of 0.231 for Case 2, suggesting reasonably good error margins given the complexity of the task. However, the low R-squared value of 0.318 may show a significant predictive capability given the multifactorial nature of disease progression (Table 4). Through the analysis on feature importance on how important each feature is for making a prediction (Figure 4), we observed the different aspects of patient data that are critical for mortality prediction and disease progression modeling. In Case 1, there were several key variables that significantly influenced the model's predictions for patient mortality. The minimum and maximum lab values are the most critical features that were most influential for predicting patient mortality, while age, admission sequence, and specific lab types were significantly contributing for making a prediction in modeling disease progression. These insights can help healthcare providers focus on the most relevant factors when assessing patient risk and tailoring interventions.

4.3 Results on Cost-Efficiency Analysis

In the analysis of cost-efficiency, the resource utilization with the usage of storage and node was evaluated. BigQuery ML leverages the power of Google Cloud's serverless data warehouse to enable ML model training directly within the BigQuery environment. This approach allows users to run machine learning algorithms using SQL queries without the need to manage any underlying infrastructure. The serverless nature of BigQuery automatically scales compute resources based on the query complexity and dataset size. This setup is particularly advantageous for users who are familiar with SQL, as it integrates seamlessly with existing data analysis workflows. In this study, BigQuery ML was used to train models

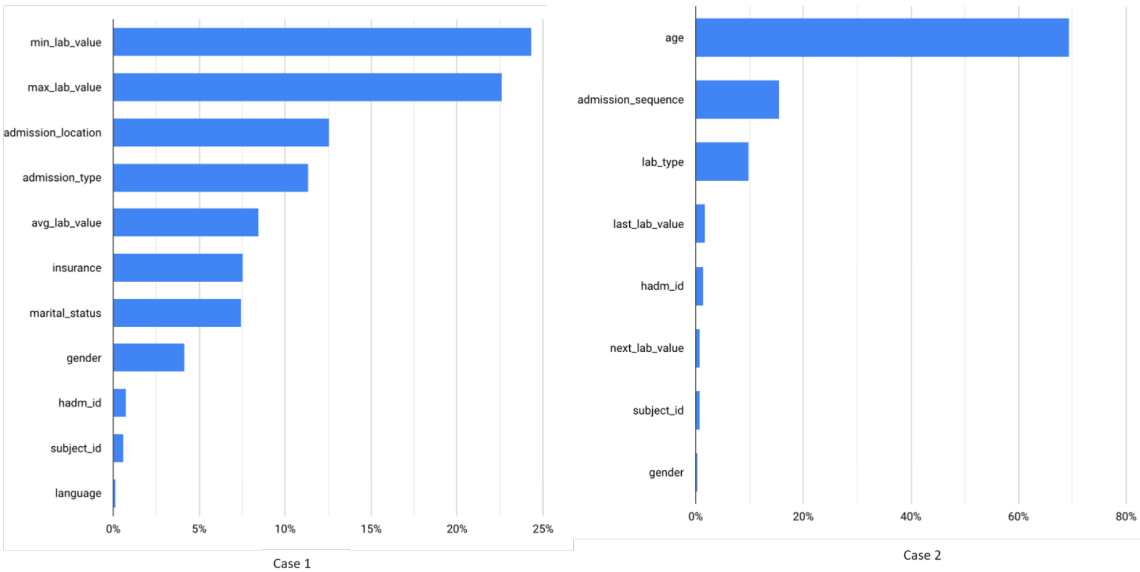

Figure 4: Feature Importance on Vertex AI.

for patient mortality prediction, utilizing its robust data handling and processing capabilities to perform these tasks efficiently and effectively [9]. Vertex AI automatically handles data preparation, model training, and hyperparameter. Vertex AI utilized AutoML for training, allocating resources based on the specified node hours. Specifically, for tabular data models, Vertex AI leverages virtual machines (VMs) optimized for machine learning tasks. This platform provides options to select the type of machine, including general-purpose, compute-optimized, and memory-optimized VMs [34]. For the experiments set for this study, each training run was allocated 1 node/hour, and the early stopping feature enabled to optimize cost-efficiency.

4.3.1 Cost on BigQuery ML. To calculate the cost involving the experiments using BigQuery, both storage and query utilizations were considered. The cost of storage utilization is based on the amount of data stored in the BigQuery data warehouse, while the cost of query processing is based on the amount of data processed during the execution of queries on the model deployment. BigQuery charges approximately $0.02 per GB for storage per month and approximately $6.25 per TB of data processed [35]. For Case 1, we stored 0.031 GB of data, resulting in a storage cost of $0.00062 per month (0.031 GB x $0.02/GB/month). The query processed 0.00003 TB of data, resulting in a query cost of $0.000189 (0.00003 TB x $6.25/TB). Combining the storage and query costs, the total cost for Case 1 is $0.000251. For Case 2, we stored 3.04 GB of data, resulting in a storage cost of $0.0608 per month (3.04 GB x $0.02/GB/month). The query processed 0.00297 TB of data, resulting in a query cost of $0.018554 (0.00297 TB x $6.25/TB). Combining these costs, the total cost for Case 2 is $0.079354. The total cost charged for each case is shown in Table 7 based on the storage utilization in each case in Table 6.

BigQuery ML seems ideal for cost-sensitive projects and smaller datasets due to its efficient resource management and lower overall costs. The cost analysis reveals that using BigQuery ML for both cases is highly cost-efficient, even with the significant difference in data sizes. For Case 1, the total monthly cost is only $0.000814, making it an extremely cost-efficient option for smaller datasets. Case 2, which involved a much larger dataset, has a total monthly cost of $0.079354. This demonstrates that BigQuery ML can handle large-scale data processing at a relatively low cost. These cost efficiencies highlight BigQuery ML's potential for scalable machine learning applications, providing cost-efficient solutions for both small and large datasets. When planning and budgeting for machine learning projects on cloud platforms, by leveraging BigQuery ML, individuals and organizations can optimize their expenses while maintaining high performance and scalability in their machine learning workflows.

Table 6: Storage Utilization Involving Experiments on BigQuery ML.

Metric	Case 1	Case 2
Total logical bytes	31.74 MB	3.04 GB
Total physical bytes	3.59 MB	402.19 MB
Number of rows	312,076	59,266,165

Table 7: Total Cost Involving Experiments on BigQuery ML.

Metric	Case 1	Case 2
Storage Cost (USD)	$0.00062/month	$0.0608/month
Query Cost (USD)	$0.000189	$0.018554
Total Cost (USD)	$0.000251	$0.079354

Table 8: Total Cost Involving Experiments on Vertex AI.

Metric	Case 1	Case 2
Budget (original)	1 node hour	1 node hour
Algorithm	AutoML	AutoML
Objective	Tabular regression	Tabular regression
Training Time (hours)	2.033 hours	2.15 hours
Cost per Node Hour (USD)	$21.252	$21.252
Total Cost (USD)	$21.25	$21.25

4.3.2 Cost on Vertex AI. Vertex AI's AutoML typically charges for training based on the node hours used. As of the most recent pricing scheme, the cost is approximately $21.252 per node hour for training tabular data [34]. For Case 1, the training time was 2.033 hours. Multiplying this by the cost per node hour, we get $42.98 (2.033 hours x $21.252/node hour). For Case 2, the training time was 2.15 hours. Multiplying this by the cost per node hour, we get $45.69 (2.15 hours x $21.252/node hour). Vertex AI offers advanced machine learning tools and flexibility but incurs higher costs due to the use of AutoML and specific resource allocations. Early stopping can optimize training duration and cost-efficiency by preventing overtraining and unnecessary computation, ensuring that resources are used more efficiently without compromising model performance. With this scheme, the total cost involving the experiments on Vertex AI is shown in Table 8. Vertex AI, while offering advanced machine learning tools and flexibility, incurred higher costs of $21.25 each case due to the use of AutoML and specific resource allocations. The cost reduction in both experiments due to early stopping highlights the feature's effectiveness in optimizing training duration and cost-efficiency [42]. By preventing overtraining and unnecessary computation, early stopping can ensure that resources are used more efficiently, reducing overall costs without compromising model performance [42]. This makes Vertex AI a cost-effective choice for machine learning tasks, especially when combined with features that optimize resource usage.

5 Discussion and Limitation

The findings of the experiments provided the implications of choosing deployment strategies between BigQuery ML and Vertex AI, based on their usability, performance and cost-efficiency. BigQuery ML's seamless integration and simple user interface make it suitable for environments where rapid data processing and ease of use are needed and ideal for users who already know SQL [36]. Vertex AI's flexibility and extensive toolsets offer advanced capabilities for complex model training, making it better suited for complex machine learning tasks where customization is key [36]. The decision on these platforms should consider the specific requirements of the application, such as the need for rapid results or the capability for detailed model tuning, which can significantly impact outcomes and operational efficiency [36]. While BigQuery ML remains the more cost-efficient with these specific scenarios, leveraging features like early stopping can enhance cost-efficiency in Vertex AI [42].

Our research has led to the development of best practices for deploying machine learning models in cloud environments. These practices are derived from our experimental findings and aim to enhance operational efficiency and user experience. It is crucial to ensure that the machine learning platform chosen integrates well with existing data processing workflows. For instance, BigQuery ML is advantageous for users familiar with SQL and existing SQL-based workflows, facilitating a smoother transition and quicker deployment. Regularly monitoring and optimizing resource usage, including storage and compute resources, helps avoid unnecessary costs. Employing practices such as early stopping in Vertex AI can significantly reduce costs without compromising model performance, as our experiments demonstrated [43]. Additionally, leveraging comprehensive documentation and training resources provided by the cloud platform can minimize the learning curve. BigQuery ML offers extensive templates and examples that make it easier for beginners to get started, thereby reducing the time and effort required to deploy machine learning models effectively [44]. These best practices ensure that organizations can efficiently deploy and manage machine learning models, maximizing both performance and cost-effectiveness.

To assist organizations in selecting the most suitable deployment architecture, we provide guidelines based on our comparative analysis of BigQuery ML and Vertex AI. It is important to assess the specific needs of the project, such as the required level of model customization, data size, and the expertise of the team. For projects requiring quick iterations and minimal configuration, BigQuery ML is suitable. For more complex tasks requiring advanced customization, Vertex AI is preferable. Cost implications of each platform must also be considered. BigQuery ML is highly cost-effective for SQL-intensive projects, whereas Vertex AI, despite its higher cost, offers advanced capabilities suitable for complex machine learning workflows. Understanding these cost dynamics is crucial for budgeting and resource allocation. Furthermore, evaluating the performance and scalability of the platform in handling large datasets is essential. BigQuery ML demonstrated efficiency in processing large-scale data with reduced subsequent iteration times, while Vertex AI provided consistent performance across iterations but at a higher computational cost [44]. This insight helps in choosing a platform that aligns with the scale and performance needs of the project [45]. By following these guidelines, organizations can make informed decisions on their cloud-based machine learning deployment strategies, ensuring optimal alignment with their project goals and resource constraints.

This study has some limitations such as potential biases from the MIMIC-IV dataset, which may not fully represent all patient demographics or clinical scenarios [36]. The ratings for documentation and learning curve presented in Table 1, are derived solely from the author's experience with the two ML-AI cloud services. Thus, these ratings may not necessarily reflect the objective views or opinions of others. The comparison of deploying models in BigQuery ML and Vertex AI reveals the challenges in directly applying and analyzing ML algorithms due to their different operational frameworks and data handling capabilities. For instance, because of these differences, the running times on the performance analysis and cost-efficiency measures may not be objectives.

6 Conclusion and Future Work

In this study, we conducted experiments to compare and analyze the usability, performance, and cost-efficiency of deploying machine learning models on Google's two ML-AI platforms: BigQuery ML and Vertex AI. The experiments used the MIMIC-IV dataset of hospitalized patients for two model deployments involving regression in each platform: predicting patient mortality and modeling disease progression. The usability analysis showed that BigQuery ML provides documents that are easy to follow, presenting a moderate learning curve for beginner cloud users. While Vertex AI offers more flexible and extensive services, it requires a deeper level of user engagement and expertise during model configurations, making it more suitable for complex ML tasks. The results also demonstrated that BigQuery ML offers a highly cost-effective solution with efficient resource management and quick deployment capabilities, particularly suitable for SQL-savvy users and large-scale data analytics tasks. Vertex AI, with its advanced AutoML and custom model training options, provided a flexible and robust environment for complex machine learning workflows, but at a higher cost. The cost analysis can be further studied as their document suggests leveraging early stopping to reduce overall cost on Vertex AI [42]. Though, Vertex AI stands out for environments where flexibility and extensive customization are necessary, accommodating a wider range of complex machine learning tasks.

Overall, the performance analysis highlighted the strengths and limitations of both BigQuery ML and Vertex AI in handling medical datasets specifically. BigQuery ML demonstrated higher initial processing time but maintained efficiency over subsequent iterations, making it more resource and cost-efficient in the long run. Vertex AI, while consistent and stable, incurred higher costs due to longer training times and specific resource allocations. The findings indicate that BigQuery ML proved to be more efficient, particularly in terms of processing time and cost, emphasizing its suitability for large-scale data processing tasks where efficiency is crucial.

These findings highlight the importance of selecting the appropriate platform based on specific task requirements, user expertise, and budget constraints, when optimizing both performance and cost-efficiency in cloud-based machine learning deployments. For cloud engineers and data scientists selecting deployment methods, it's crucial to align the choice of ML-AL platform with specific project requirements and team capabilities. They may consider BigQuery ML for projects that require quick iterations and are SQL-intensive, but Vertex AI when projects demand high customization and are managed by teams with advanced machine learning expertise.

While this study specifically focused on the performance and cost of deploying models in Google's ML-AI platforms, future studies will explore the integration of ML-AI with emerging technologies and its impact on deployment strategies in Cloud. Research can focus on:

- Developing adaptive models that can continuously learn and evolve without requiring frequent retraining.
- Enhancing model robustness to changes in data patterns, known as concept drift, remains a significant challenge in dynamic environments.
- Exploring the effectiveness of federated learning in healthcare to enhance privacy and data security while utilizing distributed data sources [39].

These directions could help in overcoming current limitations and pushing the boundaries of what is achievable with ML deployments in cloud environments. Furthermore, exploring the impact of AI in real-world settings will help bridge the gap between model performance and practical usability, ensuring that ML deployments deliver benefits in operational settings [39].

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 2334243.

References

[1] Pawar, Chandrashekhar & Ganatra, Amit & Nayak, Amit & Ramoliya, Dipak & Patel, Rajesh. 2021. Use of Machine Learning Services in Cloud. https://doi.org/10.1007/978-981-16-0965-7_5
[2] M. I. Jordan and T. M. Mitchell. 2015. Machine learning: Trends, perspectives, and prospects. Science 349, 6245 (2015), 255-260. https://doi.org/10.1126/science.aaa8415
[3] I. A. T. Hashem *et al.* 2015. The rise of 'big data' on cloud computing: Review and open research issues. Information Systems 47 (2015), 98-115. https://doi.org/10.1016/j.is.2014.07.006
[4] National Science and Technology Council. 2022. Lessons Learned from Federal Use of Cloud Computing to Support Artificial Intelligence Research and Development. Retrieved June 7, 2024 from https://www.whitehouse.gov/wp-content/uploads/2022/07/07-2022-Lessons-Learned-Cloud-for-AI-July2022.pdf
[5] Select Committee on Artificial Intelligence of the National Science & Technology Council. 2020. Recommendations for Leveraging Cloud Computing Resources for Federally Funded Artificial Intelligence Research and Development. Retrieved June 7, 2024 from https://www.nitrd.gov/pubs/Recommendations-Cloud-AI-RD-Nov2020.pdf
[6] Computing Community Consortium and AAAI (Association for the Advancement of Artificial Intelligence). 2019. A 20-Year Community Roadmap for Artificial Intelligence Research in the US. Retrieved June 7, 2024 from https://cra.org/ccc/wp-content/uploads/sites/2/2019/08/Community-Roadmap-for-AI-Research.pdf
[7] Viktoria N. Tsakalidou, Pavlina Mitsou, and George A. Papakostas. 2023. Machine Learning for Cloud Resources Management—An Overview. In Computer Networks and Inventive Communication Technologies, S. Smys, Pavel Lafata, Ram Palanisamy, and Khaled A. Kamel (Eds.). Springer Nature Singapore, Singapore, 903–915. https://doi.org/10.48550/arXiv.2101.11984
[8] D. S. Linthicum. 2017. Making Sense of AI in Public Clouds. IEEE Cloud Computing 4, 6 (Nov./Dec. 2017), 70-72. https://doi.org/10.1109/MCC.2018.1081067
[9] Rajesh Thallam. 2020. BigQuery explained: An overview of BigQuery's architecture. Retrieved June 5, 2024 from https://cloud.google.com/blog/products/data-analytics/new-blog-series-bigquery-explained-overview
[10] IBM. IaaS vs. PaaS vs. SaaS. Retrieved May 29, 2024 from https://www.ibm.com/topics/iaas-paas-saas.
[11] JavaTpoint. 2023. Introduction to Cloud Computing. Retrieved May 29, 2024 from https://www.javatpoint.com/introduction-to-cloud-computing
[12] Analytics Vidhya. 2023. Building a Machine Learning Model in BigQuery. Retrieved May 29, 2024 from https://www.analyticsvidhya.com/blog/2023/02/building-a-machine-learning-model-in-bigquery/
[13] Google Cloud Skills Boost. 2023. Build and Deploy Machine Learning Solutions with Vertex AI: Challenge Lab. Retrieved May 29, 2024 from https://www.cloudskillsboost.google/focuses/22019?parent$=$catalog
[14] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich. 2017. Data management challenges in production machine learning. In Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD), 1725-1732. https://doi.org/10.1145/3035918.3054782

[15] V. Lakshmanan, S. Robinson, and M. Munn. 2020. Machine Learning Design Patterns. O'Reilly Media, Inc. Retrieved May 29, 2024 from https://learning.oreilly.com/library/view/machine-learning-design/9781098115777/
[16] M. A. Shahid, M. M. Alam, and M. M. Su'ud. 2023. Performance Evaluation of Load-Balancing Algorithms with Different Service Broker Policies for Cloud Computing. Applied Sciences 13, 3 (2023), Article 1586. https://doi.org/10.3390/app13031586
[17] A. Chiumento, R. Bemthuis, A. Aldea, and P. Havinga. 2021. Resource Management Techniques for Cloud/Fog and Edge Computing: An Evaluation Framework and Classification. Sensors 21, 5 (2021), Article 1832. https://doi.org/10.3390/s21051832
[18] Cloudwards.net. 2024. 37 Cloud Computing Statistics, Facts & Trends for 2024. Retrieved May 29, 2024 from https://www.cloudwards.net/cloud-computing-statistics/
[19] D. S. Linthicum. 2017. Making Sense of AI in Public Clouds. IEEE Cloud Computing 4, 6 (2017), 70-72. https://doi.org/10.1109/MCC.2018.1081067
[20] M. Muniswamaiah, T. Agerwala, and C. C. Tappert. 2021. A Survey on Cloudlets, Mobile Edge, and Fog Computing. In Proceedings of the 8th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud) / 7th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom), 139-142. https://doi.org/10.1109/CSCloud-EdgeCom52276.2021.00034
[21] J. Vemulapati, A. S. Khastgir, and C. Savalgi. 2019. AI Based Performance Benchmarking & Analysis of Big Data and Cloud Powered Applications: An in Depth View. In Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering (ICPE '19), 103-109. https://doi.org/10.1145/3297663.3309676
[22] Turing. 2023. 5 Key Trends in Cloud Computing in 2023. Retrieved May 29, 2024 from https://www.turing.com/blog/cloud-computing-trends/
[23] X. Li, Y. Li, T. Liu, J. Qiu and F. Wang. 2009. The Method and Tool of Cost Analysis for Cloud Computing. In Proceedings of the 2009 IEEE International Conference on Cloud Computing, 93-100. https://doi.org/10.1109/CLOUD.2009.84
[24] A. Qayyum, A. Ijaz, M. Usama, W. Iqbal, J. Qadir, Y. Elkhatib, and A. Al-Fuqaha. 2020. Securing Machine Learning in the Cloud: A Systematic Review of Cloud Machine Learning Security. Frontiers in Big Data 3 (2020). https://www.frontiersin.org/articles/10.3389/fdata.2020.587139/full
[25] Hindawi. 2021. Current Research Trends in IoT Security: A Systematic Mapping Study. Retrieved May 29, 2024 from https://www.hindawi.com/journals/misy/2021/8847099/
[26] Analytics Vidhya. 2023. Building a Machine Learning Model in BigQuery. Retrieved May 29, 2024 from https://www.analyticsvidhya.com/blog/2023/02/building-a-machine-learning-model-in-bigquery/
[27] W. Kambale. 2023. Train ML Model in SQL with BigQuery. Retrieved May 29, 2024 from https://kambale.dev/ml-in-sql
[28] Google Cloud Press Corner. 2023. AI21 Labs Collaborates with Google Cloud to Integrate Generative AI Capabilities with BigQuery. Retrieved May 29, 2024 from https://www.googlecloudpresscorner.com/2023-08-29-AI21-Labs-Collaborates-with-Google-Cloud-to-Integrate-Generative-AI-Capabilities-with-BigQuery
[29] MIMIC-IV Documentation. Retrieved June 7, 2024 from https://mimic.mit.edu/docs/iv/
[30] Google Cloud Skills Boost. 2023. Building Demand Forecasting with BigQuery ML. Retrieved May 29, 2024 from https://www.cloudskillsboost.google/focuses/16547?parent$=$catalog
[31] Analytics Vidhya. 2023. Introduction to BigQuery ML. Retrieved May 29, 2024 from https://www.analyticsvidhya.com/blog/2023/01/introduction-to-bigquery-ml/
[32] D. W. Hosmer, S. Lemeshow, and R. X. Sturdivant. 2013. Introduction to the logistic regression model. In Applied Logistic Regression, 3rd ed. Wiley, New York, NY, USA, 1-35. Retrieved May 29, 2024 from https://synapse.koreamed.org/upload/synapsedata/pdfdata/0006jkan/jkan-43-154.pdf
[33] Google Cloud. 2024. Tabular Data Overview | Vertex AI. Retrieved June 5, 2024 from https://cloud.google.com/vertex-ai/docs/tabular-data/overview
[34] Google Cloud. 2024. Vertex AI pricing. Retrieved June 5, 2024 from https://cloud.google.com/vertex-ai/pricing#tabular-data
[35] Google Cloud. 2024. BigQuery pricing. Retrieved June 5, 2024 from https://cloud.google.com/bigquery/pricing#bigquery-pricing
[36] L. A. Celi, J. Cellini, M.-L. Charpignon, E. C. Dee, F. Dernoncourt, R. Eber, *et al.* 2022. Sources of bias in artificial intelligence that perpetuate healthcare disparities—A global review. PLOS Digital Health 1, 3 (2022), Article e0000022. https://journals.plos.org/digitalhealth/article?id$=$10.1371/journal.pdig.0000022
[37] C. J. Kelly, A. Karthikesalingam, M. Suleyman, G. Corrado, and D. King. 2019. Key challenges for delivering clinical impact with artificial intelligence. BMC Medicine 17, Article 195 (2019). https://bmcmedicine.biomedcentral.com/articles/10.1186/s12916-019-1426-2
[38] Zitnik Lab. 2020. Interpretable Machine Learning in Healthcare. Retrieved May 29, 2024 from https://zitniklab.hms.harvard.edu/publications/papers/interpretableML-survey20.pdf
[39] Machine Learning Models. 2023. Scaling ML Model Deployment: Best Practices and Strategies. Retrieved May 29, 2024 from https://machinelearningmodels.org/scaling-ml-model-deployment-best-practices-and-strategies/.
[40] ISO 9241-221:2023. https://www.iso.org/standard/81384.html.
[41] A. Abraham and J. Yang, "A Comparative Analysis of Performance and Usability on Serverless and Server-Based Google Cloud Services," 2023 Springer Book Series: Lecture Notes in Networks and Systems, https://doi.org/10.1007/978-3-031-33743-7_33.
[42] Google Cloud Community. 2024. Time to Train a New Model in Vertex AI. Retrieved June 9, 2024 from https://www.googlecloudcommunity.com/gc/AI-ML/Time-to-train-a-new-model-in-Vertex-AI/m-p/611730
[43] G. Senthilkumar, K. Tamilarasi, N. Velmurugan, and J. K. Periasamy. 2023. Resource Allocation in Cloud Computing. Journal of Advances in Information Technology 14, 5 (2023), 1063-1072. https://www.jait.us/show-232-1416-1.html
[44] S. I. Ele, U. R. Alo, H. F. Nweke, and O. A. Ofem. 2023. Regression-Based Machine Learning Framework for Customer Churn Prediction in Telecommunication Industry. Journal of Advances in Information Technology 14, 5 (2023), 1046-1055. https://www.jait.us/show-232-1414-1.html
[45] Trong-Minh Hoang, Trang-Linh Le Thi, and Nguyen Minh Quy. 2023. A Novel Distributed Machine Learning Model to Detect Attacks on Edge Computing Network. Journal of Advances in Information Technology 14, 1 (2023), 153-159. https://www.jait.us/show-224-1302-1.html.