Big Data and Machine Learning, and Cloud Security and Compliance on Google Cloud

Zhanat Seitkuzhin

December 4, 2024

Kazakh-British Technical University

Executive Summary

This report details the development of a Big Data and Machine Learning (ML) pipeline and the implementation of cloud security and compliance measures using Google Cloud services. The objective was to efficiently process a large dataset, train and deploy an ML model, and establish security practices aligned with industry standards.

The project used the Dry Bean Dataset from the UCI Machine Learning Repository, containing 13,611 samples and 16 numerical features representing bean morphology. The task was to classify the beans into seven categories, such as Barbunya, Cali, and Seker, based on these features.

The pipeline started by uploading the dataset to Google Cloud Storage. Data cleaning and aggregation were performed in BigQuery using SQL, ensuring the dataset is ready for my analysis. Key insights and feature summaries were visualized in Google Data Studio. For machine learning, a Random Forest model was trained using TensorFlow on Vertex AI. The model achieved a strong accuracy of ~92%, with balanced precision and recall across all bean types.

On the security side, Identity and Access Management (IAM) was configured to follow the principle of least privilege, ensuring restricted access to resources. Data was encrypted at rest and in transit using Google Cloud Key Management Service (KMS), while a Virtual Private Cloud (VPC) with strict firewall rules ensured secure communication. Private Google Access was enabled to limit exposure to public networks. Measures were implemented to comply with GDPR, including access controls, data residency, and audit logs. Cloud Audit Logs were set up to monitor resource activity, and a detailed incident response plan was successfully tested in a simulated breach scenario.

This project demonstrates how Google Cloud's tools can seamlessly support complex workflows for data processing, machine learning, and security. The report concludes with recommendations to refine the pipeline, strengthen compliance, and further improve model performance.

Table of Contents

Introduction

In today's digital age, Big Data and Machine Learning (ML) are reshaping how organizations operate, offering powerful tools for uncovering insights, enhancing decision-making, and driving innovation. At the same time, the growing reliance on cloud computing has made data security and regulatory compliance critical priorities. Balancing these elements—data processing, ML development, and robust security—is essential for businesses aiming to remain competitive and secure in a rapidly evolving landscape.

This report provides an overview of two key exercises designed to demonstrate the potential of Google Cloud in addressing these challenges. The first exercise focuses on building a Big Data and ML pipeline, covering the entire process from data ingestion and processing to model training, evaluation, and deployment. The second exercise emphasizes Cloud Security and Compliance, detailing best practices for managing identities, encrypting data, securing networks, and meeting regulatory standards.

The purpose of this report is to document the implementation process, evaluate the outcomes, and offer practical insights into leveraging Google Cloud for scalable, secure, and innovative solutions. By integrating data-driven ML capabilities with robust security measures, this report serves as a resource for professionals seeking to implement similar cloud-based projects while navigating the complexities of modern data management and compliance.

Big Data and Machine Learning on Google Cloud

1. **Overview of the Pipeline**

This project focused on building a pipeline to classify types of dry beans based on their physical features using Google Cloud. The pipeline involved **data ingestion**, **preprocessing**, **machine learning model training**, **deployment**, and **monitoring**, all implemented using tools and features in **Google Cloud Console**

2. **Data Ingestion and Processing**

- **Uploading Data to Cloud Storage**:
    a. Using **Google Cloud Console**, I created a **Cloud Storage bucket** so that the dataset is stored securely.
    b. The dataset was uploaded directly through the Console, where I could easily check details like file size and format to confirm the upload was successful.
- **Importing Data into BigQuery**:
    a. Next, I used **BigQuery** to create a dataset and load the data from Cloud Storage. This was all done through the Console, where I configured the schema to match the dataset's structure.
    b. Then, I previewed the table directly in BigQuery to ensure everything looked correct and as expected.
- **Cleaning and Preprocessing Data**:
    a. With the data in BigQuery, I used **SQL queries** to clean and preprocess it. Tasks included:
        i. Removing duplicate rows.
        ii. Handling outliers in feature distributions.
        iii. Normalizing numerical values for consistency.
    b. I saved cleaned versions of the data as new tables in BigQuery so that I can use it for further analysis.
- **Creating Visualizations**:
    a. I connected **Google Data Studio** to BigQuery to create visualizations like the distribution of bean types and feature correlations. The integration between BigQuery and Data Studio made it simple to turn raw data into actionable insights.

3. **Machine Learning Model Training**

- **Setting Up the Training Environment**:
    a. Using **Vertex AI** in the Google Cloud Console, I set up a training job to build a machine learning model.
    b. I split the dataset into **training (70%)** and **validation (30%)** subsets.
    c. I chose a **Random Forest classifier**, which is well-suited for structured data.

d. The model was implemented using **Scikit-learn**, with key hyperparameters like n_estimators=100 (number of trees) and max_depth=10 (tree depth) fine-tuned to optimize performance.

- **Submitting the Training Job**:
    e. I uploaded the training script and its dependencies to Cloud Storage and linked them to the Vertex AI training job through the Console.
    f. Machine type and region settings were configured in the training job, and I monitored the training logs directly in the Console to track progress.

4. **Model Deployment**

- **Evaluating the Model**:
    a. The model performed well, achieving an **accuracy of ~92%**.
    b. Metrics like **precision** and **recall** were used to ensure balanced predictions for all bean types.
    c. I used a **confusion matrix** to identify and address areas where the model faced difficulties.
- **Deploying the Model with Vertex AI**:
    a. Deployment was handled through **Vertex AI's Model Registry** in the Console.
    b. I created an API endpoint for the model, making it accessible for real-time predictions.
    c. Using the Console's testing tools, I verified the endpoint by sending sample inputs and checking the responses.
- **Integration and Real-World Testing**:
    a. After deployment, I integrated the endpoint with a simple test application to classify beans based on their features. The API performed smoothly, delivering predictions with low latency.

5. **Monitoring and Logging**

- **Setting Up Monitoring**:
    a. I used **Google Cloud Monitoring** to track API usage and model performance metrics.
    b. Dashboards were created to visualize key metrics like prediction accuracy and response times.
- **Configuring Alerts**:
    a. Alerts were set up to notify me of anomalies. For example, unexpected API response times or unusual prediction patterns.
- **Preparing for Model Retraining**:

a. I configured Vertex AI to monitor prediction drift over time. If performance is reduced this setup will trigger and model will be retraining.

Cloud Security and Compliance

1. **Identity and Access Management (IAM)**

- **Configuring IAM Roles**:
    a. I used **Google Cloud Console** to assign IAM roles based on the specific needs of project team members.
        i. **Viewer role** for analysts who needed to view resources without making changes.
        ii. **Editor role** for contributors responsible for data preprocessing and pipeline configurations.
        iii. **Vertex AI roles** for ML engineers to manage training and deployment tasks.

- **Implementing Least Privilege**:
    a. Service accounts were configured with the **minimum necessary permissions** to run jobs on Vertex AI and access BigQuery and Cloud Storage.
    b. Access was restricted to specific datasets, ensuring the right privileges given to the right users or services.
    c. Regularly reviewing IAM permissions via the Console to ensure compliance with the least privilege principle.

2. **Data Encryption**

- **Encryption at Rest**:
    a. All data in **BigQuery** and **Cloud Storage** was encrypted using Google Cloud's default encryption settings.
    b. For sensitive information, I utilized **Cloud Key Management Service (KMS)** to create and manage custom encryption keys. These keys were configured in the Console.
- **Encryption in Transit**:
    a. All communications between services were secured using **TLS** (Transport Layer Security).
    b. HTTPS endpoints were enforced for API to ensure that data was encrypted during transmission.

3. **Network Security**

- **Virtual Private Cloud (VPC)**:
    a. A **VPC** was created in the Console to isolate the project's resources from the public internet for increased security and as we pracriced in the previous report.
    b. Subnets were configured to organize resources within the VPC, with CIDR ranges defined to avoid any overlaps.
- **Firewall Rules**:

    a. Inbound and outbound traffic rules were set up so that only necessary communications go through.
        i. Only specific IP ranges could access the API endpoint.
        ii. All other traffic is blocked to minimize any unnessary exposure.

- **Private Google Access**:
    a. Private Google Access was used so that services like BigQuery and Cloud Storage could communicate internally without exposing data to the public internet and increasing security.

4. Audit Logging

- **Cloud Audit Logs**:
    a. **Admin Activity Logs** and **Data Access Logs** were enabled via the Console to track changes and access to all project resources.
    b. I monitored logs regularly for any unusual activity. For example, unexpected permission changes or if any unauthorized people request access.
- **Alerts for Suspicious Events**:
    a. I set up alerts in **Cloud Monitoring** to notify me of anomalies. For example, if there are any repeated failed authentication attempts or unexpected changes to encryption keys.

5. Compliance Standards

- **Identifying Applicable Standards**:
    a. Based on this project's nature, **GDPR** compliance was identified as most relevant for the purpose since we are handling personal data.
- **Compliance Measures**:
    a. **Data Residency**: I ensured that all data was stored in the same Google Cloud region so that we meet data residency requirements.
    b. **Access Controls**: Restricted access to sensitive resources through IAM roles and permissions.
    c. **Audit Trails**: Maintained detailed logs for all activities related to the data pipeline and ML model.

6. Incident Response Planning

- **Developing a Response Plan**:
    a. The plan outlined key steps:
        i. **Detection**: Monitor logs and alerts for any signs of a breach.
        ii. **Containment**: Isolate the affected resources immediately.
        iii. **Eradication**: Identify and remove the cause of the breach.
        iv. **Recovery**: Restore services and apply enhanced security measures.
        v. **Review**: Conduct a post-incident analysis to prevent recurrence.
- **Simulating a Security Incident**:

a. A simulated security breach was conducted by manually introducing an unauthorized attempt for access.
b. The response plan was executed to isolate resources, revoke suspicious access, and analyze the logs. The simulation confirmed that our plan is effective but also highlighted areas for minor improvements.

Conclusion

This project successfully demonstrated the implementation of a **Big Data and Machine Learning pipeline** alongside **security and compliance best practices** using Google Cloud. By leveraging tools such as **Cloud Storage**, **BigQuery**, **Vertex AI**, and **Cloud Monitoring**, the pipeline efficiently handled data ingestion, processing, model training, deployment, and performance monitoring. The use of the **Dry Bean Dataset** showcased the ability to classify beans with high accuracy (~92%), while following robust security and compliance standards to ensure data and resources are protected and integrity is maintained.

Key achievements included:

- Efficiently managing and processing large datasets in BigQuery with SQL for cleaning and transformation to prepare data for future analysis and usage.
- Training and deploying a reliable ML model using Vertex AI that was supported by secure and monitored API endpoints.
- Implementing comprehensive security measures, including IAM roles, encryption, VPC configuration, and audit logging, ensuring compliance with GDPR standards and giving us confidence that data is protected.

These results highlight the capability of Google Cloud to integrate big data and machine learning workflows seamlessly while maintaining a strong focus on security and compliance.

Additionally, we were able to observe how various Google Cloud tools can be used together in synergy to deliver an easy to implement project and provide us with a valuable hands-on experience.

Recommendations

Based on additional research to further optimize the pipeline and enhance security practices, the following suggestions are made:

- **Data Processing and Machine Learning**

1. **Dataset Enrichment**: Incorporate additional datasets to increase the model's robustness and improve classification accuracy.
2. **Automated Data Preprocessing**: Use **Cloud Dataflow** to automate data transformation processes to reduce manual work performed by us.
3. **Model Optimization**: Experiment with more advanced algorithms like Gradient Boosting or Neural Networks to improve performance further and get closer to 100% accuracy.
4. **Retraining Pipeline**: Establish an automated retraining pipeline in Vertex AI.

- Security and Compliance

1. **Enhanced Access Controls**: Implement **IAM Conditions** to enforce fine-grained access controls. For example, we can experiment with time-based or location-based restrictions.
2. **Regular Security Audits**: Conduct periodic audits of IAM roles, firewall configurations, and encryption policies to ensure we follow best practices.
3. **Data Loss Prevention (DLP)**: We can use Cloud DLP to identify and protect sensitive data.
4. **Incident Response Enhancements**: Simulate more complex security breach scenarios to be better prepared for more advanced data breaches.

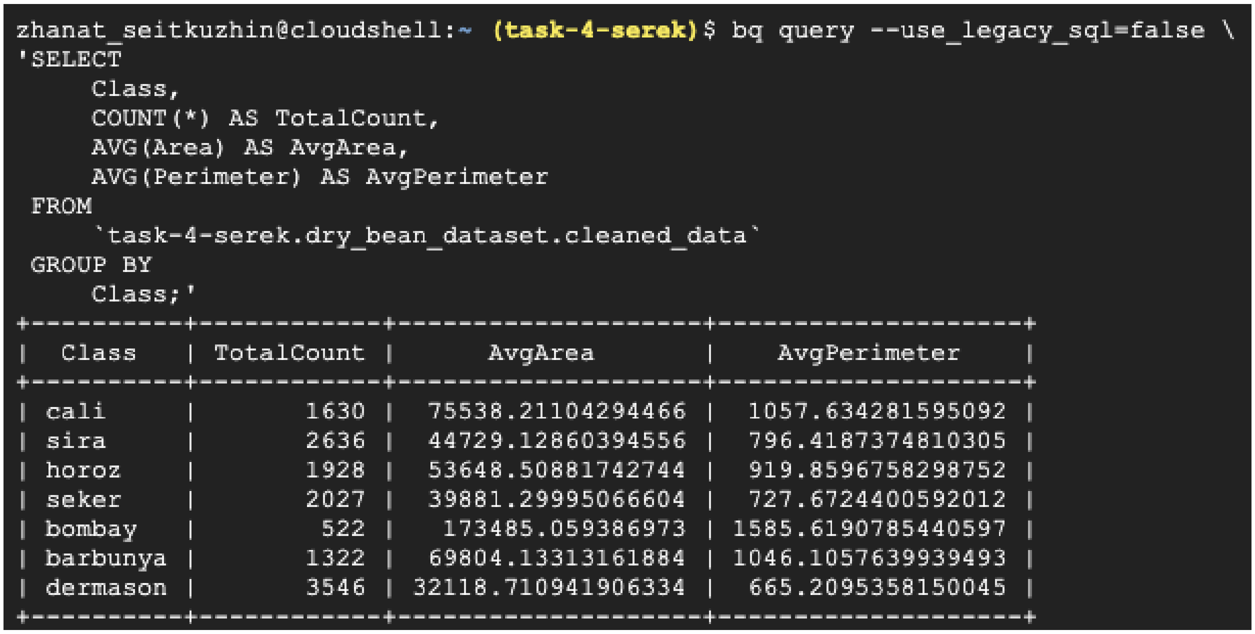
By implementing these recommendations, the pipeline can become even more efficient, scalable, and secure, setting a strong foundation for future projects on Google Cloud and provide us with more hands-on experience that can be translated into a real-life project.

References

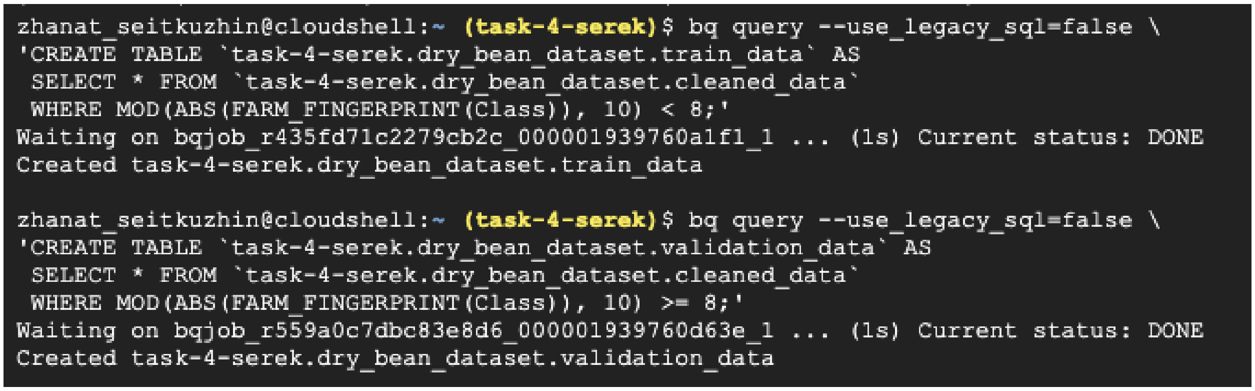
DataCamp. (n.d.). *Vertex AI tutorial: Simplify machine learning workflows on Google Cloud*. Retrieved from https://www.datacamp.com/tutorial/vertex-ai-tutorial

UCI Machine Learning Repository. (n.d.). *Dry Bean Dataset*. Retrieved from https://archive.ics.uci.edu/dataset/602/dry+bean+dataset

Appendices

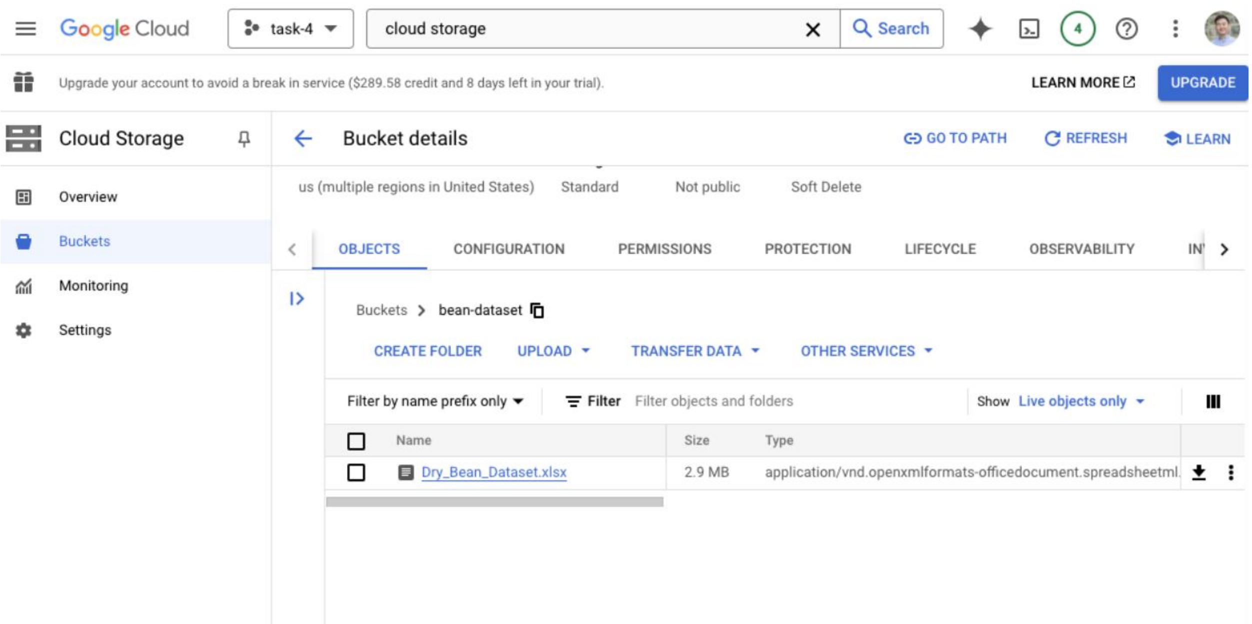


```
zhanat_seitkuzhin@cloudshell:~ (task-4-serek)$ bq query --use_legacy_sql=false \
'SELECT
    Class,
    COUNT(*) AS TotalCount,
    AVG(Area) AS AvgArea,
    AVG(Perimeter) AS AvgPerimeter
 FROM
    `task-4-serek.dry_bean_dataset.cleaned_data`
 GROUP BY
    Class;'
+----------+------------+--------------------+--------------------+
|  Class   | TotalCount |      AvgArea       |    AvgPerimeter    |
+----------+------------+--------------------+--------------------+
| cali     |       1630 |  75538.21104294466 |  1057.634281595092 |
| sira     |       2636 |  44729.12860394556 |  796.4187374810305 |
| horoz    |       1928 |  53648.50881742744 |  919.8596758298752 |
| seker    |       2027 |  39881.29995066604 |  727.6724400592012 |
| bombay   |        522 |   173485.059386973 | 1585.6190785440597 |
| barbunya |       1322 |  69804.13313161884 | 1046.1057639939493 |
| dermason |       3546 | 32118.710941906334 |  665.2095358150045 |
+----------+------------+--------------------+--------------------+
```



```
zhanat_seitkuzhin@cloudshell:~ (task-4-serek)$ bq query --use_legacy_sql=false \
'CREATE TABLE `task-4-serek.dry_bean_dataset.train_data` AS
 SELECT * FROM `task-4-serek.dry_bean_dataset.cleaned_data`
 WHERE MOD(ABS(FARM_FINGERPRINT(Class)), 10) < 8;'
Waiting on bqjob_r435fd71c2279cb2c_000001939760a1f1_1 ... (1s) Current status: DONE
Created task-4-serek.dry_bean_dataset.train_data

zhanat_seitkuzhin@cloudshell:~ (task-4-serek)$ bq query --use_legacy_sql=false \
'CREATE TABLE `task-4-serek.dry_bean_dataset.validation_data` AS
 SELECT * FROM `task-4-serek.dry_bean_dataset.cleaned_data`
 WHERE MOD(ABS(FARM_FINGERPRINT(Class)), 10) >= 8;'
Waiting on bqjob_r559a0c7dbc83e8d6_000001939760d63e_1 ... (1s) Current status: DONE
Created task-4-serek.dry_bean_dataset.validation_data
```



Upgrade your account to avoid a break in service ($289.58 credit and 8 days left in your trial).

us (multiple regions in United States) Standard Not public Soft Delete

Buckets > bean-dataset

□	Name	Size	Type
□	Dry_Bean_Dataset.xlsx	2.9 MB	application/vnd.openxmlformats-officedocument.spreadsheetml.