KAZAKH BRITISH TECHNICAL UNIVERSITY

REPORT

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

Akhmetkan Azhar

05.12.2024

Table of Contents

Executive summary

This report details the results of the big data, machine learning, and security projects implemented in the Google Cloud environment. The main objective of the work was to demonstrate Google Cloud capabilities for data processing, training machine learning models, deploying them, and applying best practices in data security and compliance with regulatory standards.

The project included the following key stages:

1. **Setting up the project in Google Cloud:** creating a project, enabling the necessary APIs, and defining the environment for work.
2. **Data collection and loading:** using a large dataset, storing it in Google Cloud Storage, and preparing it for processing.
3. **Data processing:** analyzing and cleaning the data using BigQuery, as well as generating visualizations to understand its structure.
4. **Training the machine learning model:** developing and tuning the model using TensorFlow and Scikit-learn, including working with hyperparameters and splitting the data into training and validation sets.
5. **Deploying the model:** publishing the model using the AI Platform capabilities to provide access via API.
6. **Monitoring and security:** setting up logging mechanisms, model monitoring, and data protection measures.

This project was a significant step in exploring the practical possibilities of cloud technologies for implementing complex data analysis tasks and ensuring their security.

Introduction

In the modern world, data processing and machine learning technologies are the basis for many business processes and scientific research. Cloud platforms such as Google Cloud provide unique opportunities for efficient work with large volumes of data, developing intelligent models and ensuring their security.

This report describes the stages of creating a data processing channel, training a machine learning model and implementing measures to ensure data security. The main objectives of the project:

- Implementation of the full life cycle of data processing and machine learning.
- Implementation of cloud security standards, such as access control, encryption and monitoring.
- Demonstration of project compliance with regulatory requirements, including GDPR and HIPAA.

This report is intended for professionals interested in the integration of machine learning and security in a cloud environment.

Big Data and Machine Learning on Google Cloud

Overview of the Pipeline

This project implemented a complete data processing and machine learning pipeline on the Google Cloud platform. The pipeline includes data collection, uploading to cloud storage, processing using BigQuery, training a machine learning model, and deploying the model for use in production. Particular attention is paid to monitoring and logging to ensure stability and performance control. To go to BigQuery you first need to go to your Google Cloud account, where the main page will open as shown in Figure 1.1.

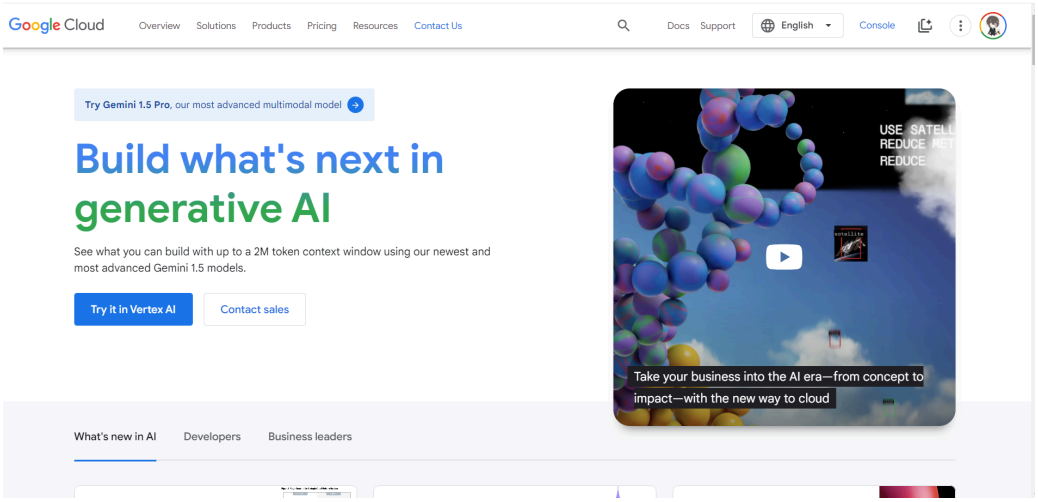

Figure 1.1. Google Cloud's home page

In the upper right corner you will see a button called "Console", which is shown in Figure 1.2, which will move your page to the Google Cloud console.

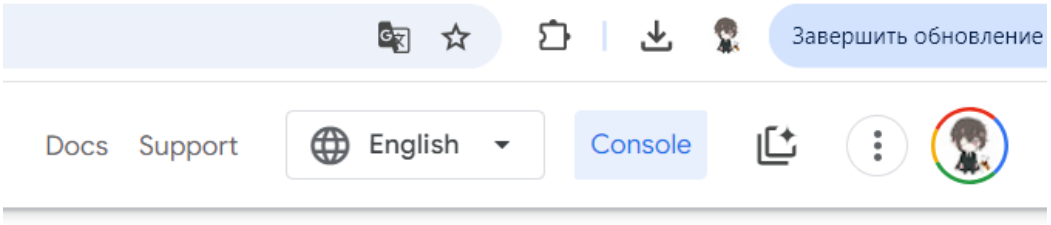

Figure 1.2. "Console" button

The main page will open in front of you with many features of the Google Cloud Console. However, we will need to find the "BigQuery" icon among them to start working with it, which is shown in Figure 1.3.

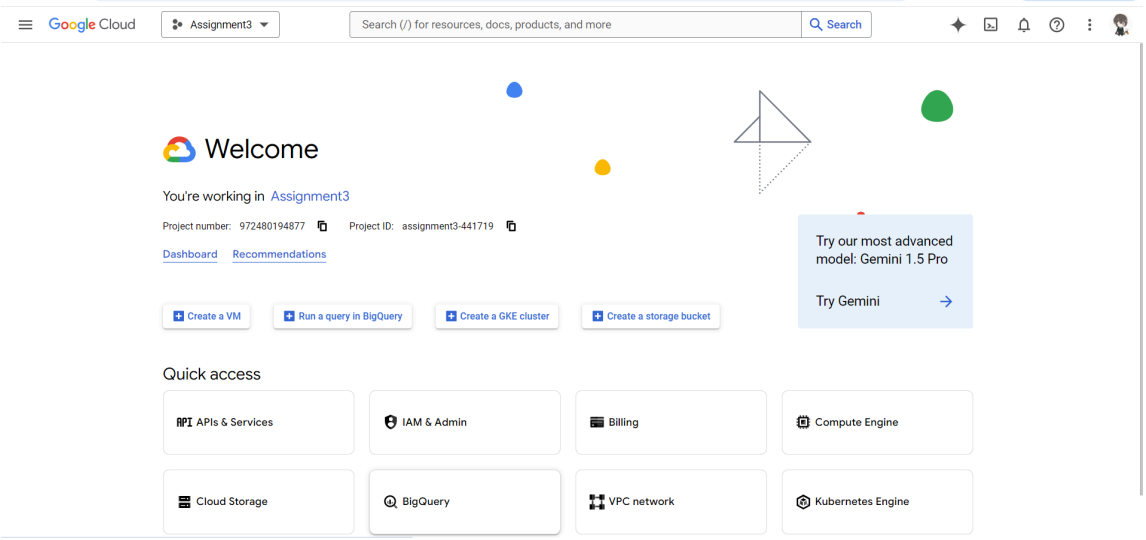

Figure 1.3. Google Cloud Console's home page

In the opened BigQuery's window shown in Figure 1.4, you can start working with the task.

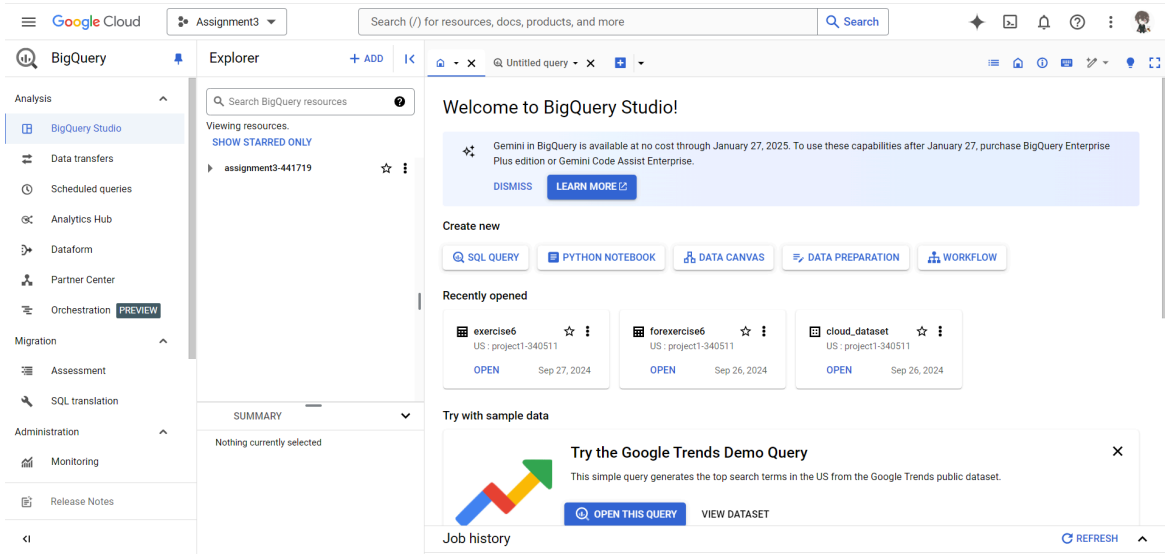

Figure 1.4. BigQuery's home page

Data Ingestion and Processing

The dataset used in the project contains physicochemical characteristics and quality assessments of Portuguese Vinho Verde wines. It consists of 12 variables:

1. Physicochemical characteristics (input variables):
    - fixed acidity
    - volatile acidity
    - citric acid
    - residual sugar
    - chlorides
    - free sulfur dioxide
    - total sulfur dioxide
    - density
    - pH

- sulphates
- alcohol

2. Quality (target variable):
    - quality — wine quality assessment on a scale from 0 to 10.

Purpose of using the dataset is transform the problem into a binary classification: divide wines into "good" (score above 6.5) and "bad".

To build a machine learning model to predict wine quality based on its physicochemical characteristics.

To carry out the wine quality data analysis project, all actions were performed on the Google Cloud platform. First, a project was created in the Google Cloud Console in Figure 1.1. In opened window as in Figure 1.3 to do exercise, the "Select a project" menu was selected at the top of the console, where the "New Project" button was pressed as in Figure 1.5.

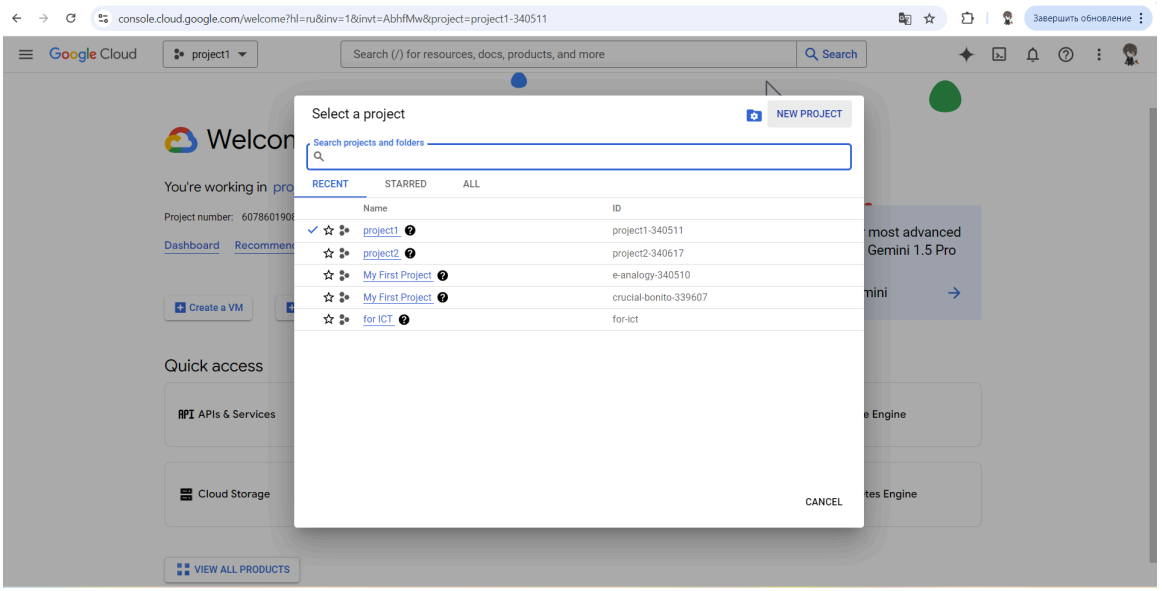

Figure 1.5 List of projects

A window will open where you will need to give a name to the new project and make some settings on how to specify the organization and click “CREATE” button for creation, which figured in Figure 1.7.

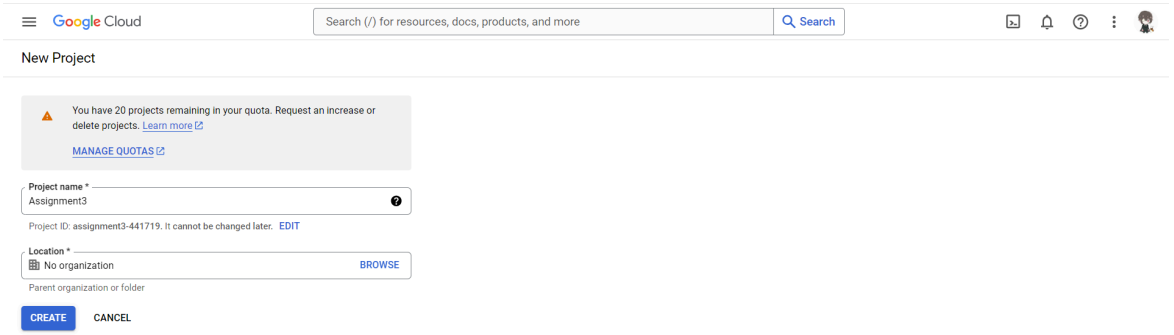

Figure 1.7 Project creation window

After creating the project, as showed in Figure 1.8 the necessary APIs were enabled via "Navigation Menu" → "APIs & Services" → "Library" in Figure 1.9.

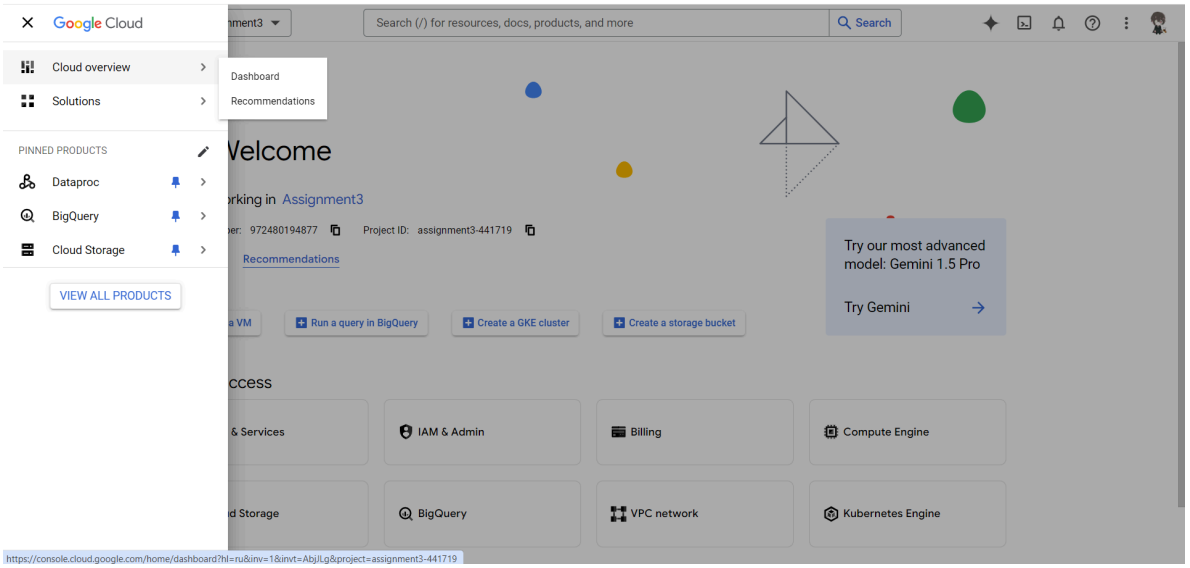

Figure 1.8 Navigation Menu

Here, the BigQuery API, Cloud Storage API, and AI Platform API are activated, which are necessary for working with data and training the model.

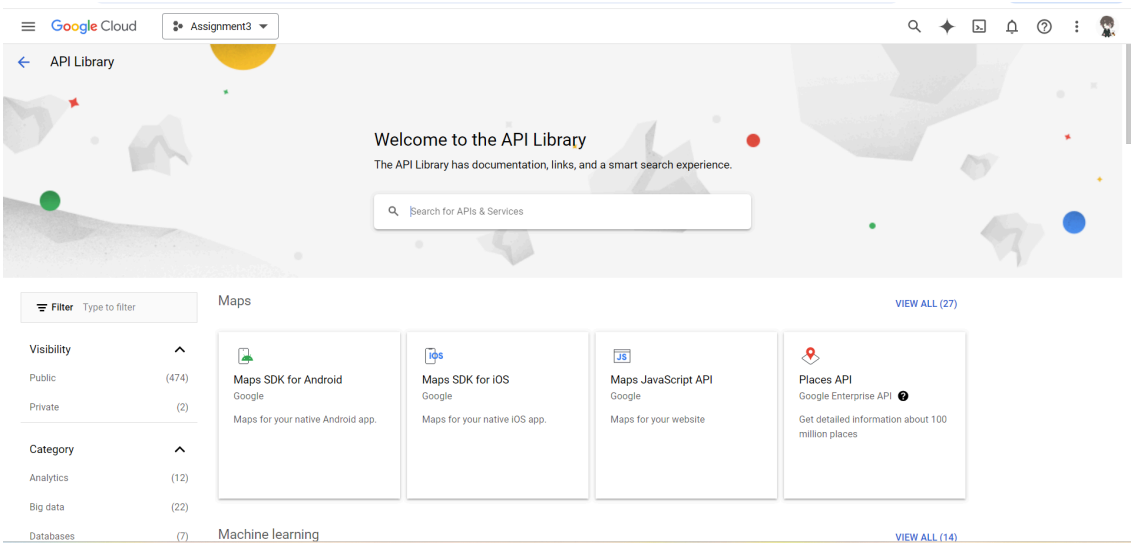

Figure 1.9 Library

Next, Google Cloud Storage was configured to store data. In the "Navigation Menu" menu, "Storage" → "Browser" was selected, where a new bucket was created as in Figure 2.1.

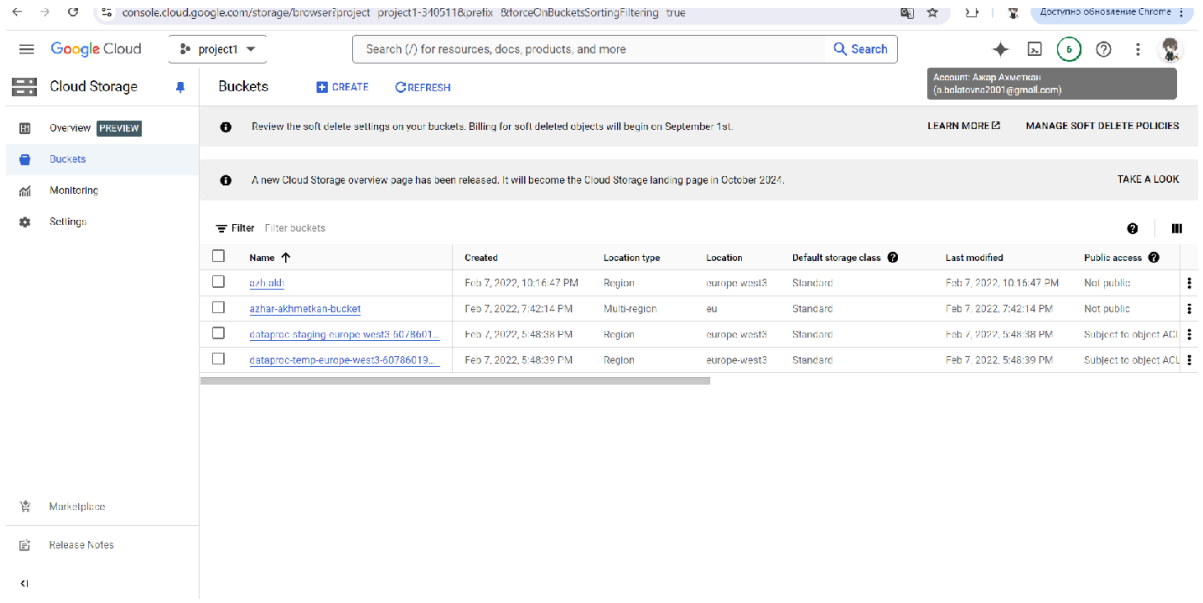

Figure 2.1 Bucket creation

The bucket settings specify the region "US-central" and the storage level "Standard". After the bucket was created, the original file winequality-red.csv was uploaded from the local computer via the "Upload Files" button inside the bucket as in Figure 2.2.

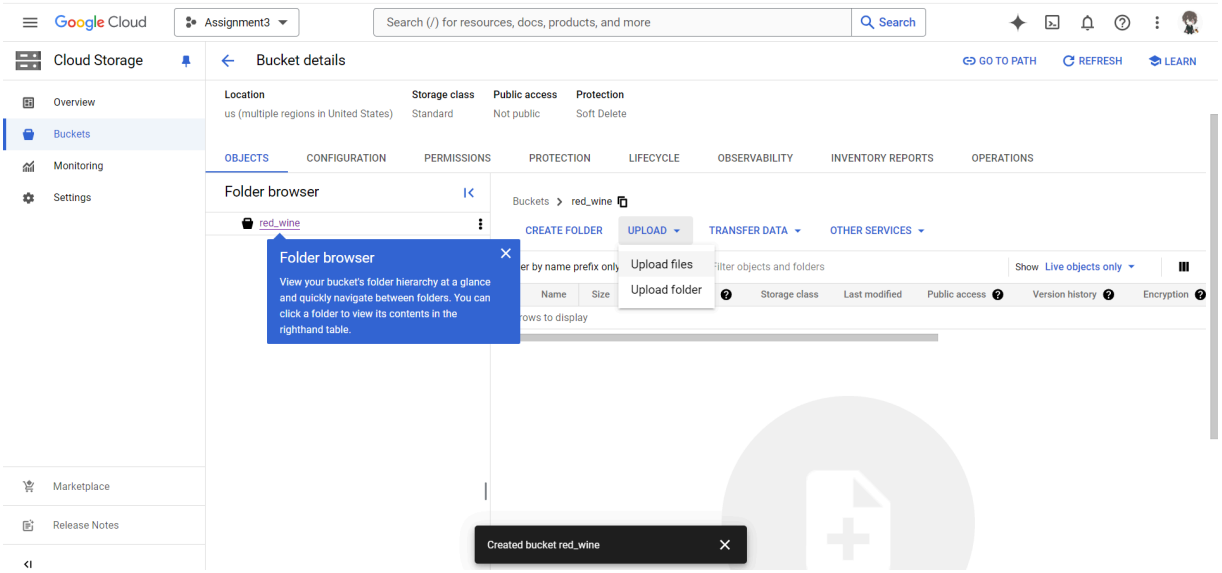

Figure 2.2 Upload File

For further analysis, the data from the bucket was imported into BigQuery. First, a new wine_quality dataset was created in BigQuery by selecting "Create Dataset" in the left panel of BigQuery as in Figure 2.3.

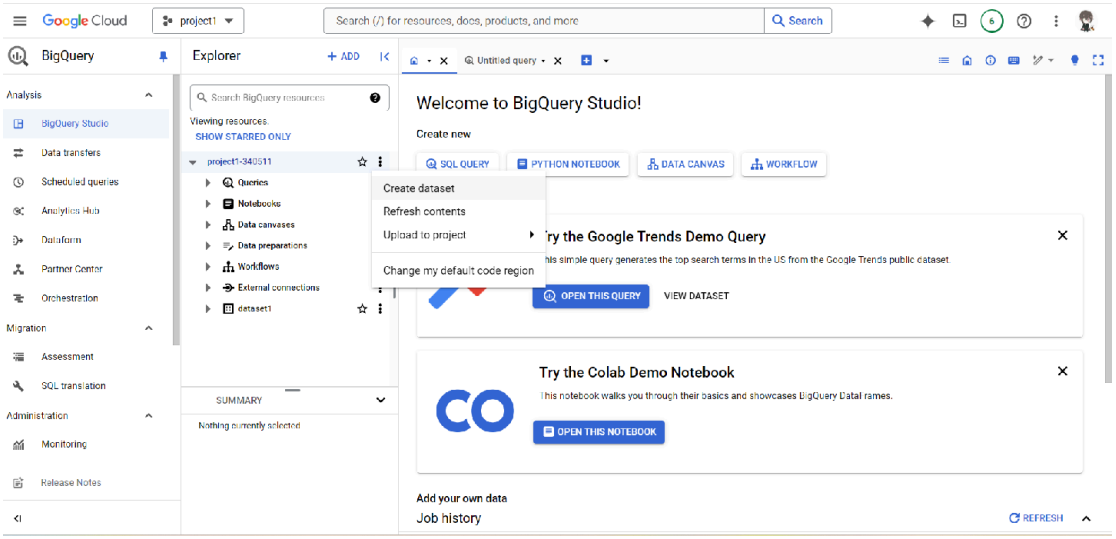

Figure 2.3 Create Dataset

After that, a table was created via "Create Table", where "Google Cloud Storage" was selected as the data source in Figure 2.4. The path to the file was specified in the format gs://wine-quality-dataset/winequality-red.csv, and for ease of use, the "Auto detect" option was enabled to automatically determine the data scheme.

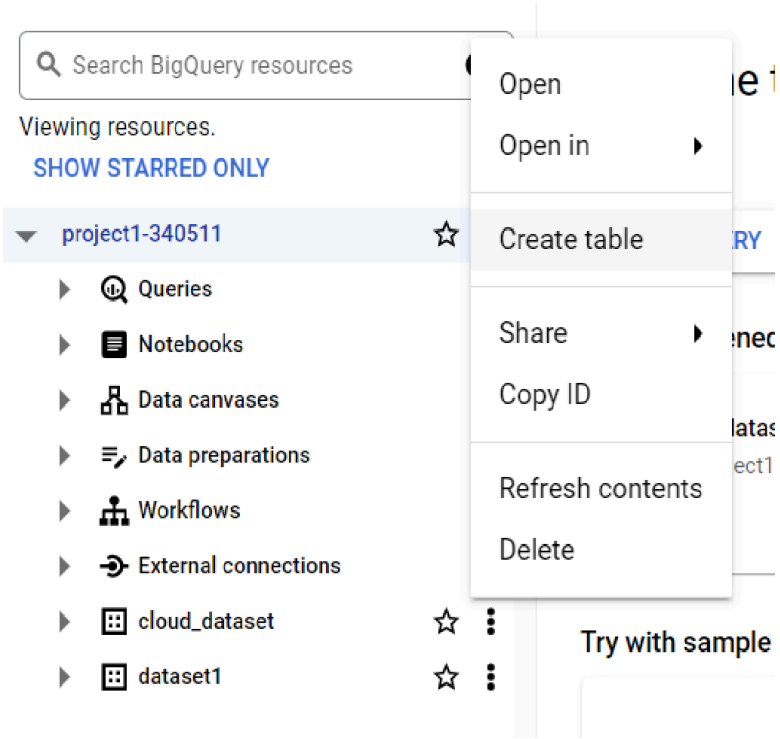

Figure 2.4 Create Table

After loading the data, processing began in BigQuery. SQL queries were executed via the "Query Table" available in the table interface. To clean the data, rows with missing values were removed, and the target variable quality was converted into a category. Values above 6.5 were classified as "good", the rest as "not good", using the SQL query:

```
SELECT *,
              CASE WHEN quality > 6.5 THEN 'good'
               ELSE 'not good' END AS quality_label
FROM `wine_quality.table`
```

Machine Learning Model Training

After preparing the data in BigQuery and exporting it, a machine learning model was built using Python. For this task, the Scikit-learn library and the Random Forest algorithm, which is suitable for classification, were used.

The data was split into training and testing sets (e.g. 75% for training and 25% for testing).

```
from sklearn.model_selection import train_test_split

# Convert the 'quality' column into a binary classification problem
# Wines with quality > 6.5 are classified as 'good' (1), else 'not good'
(0)
updated_wine_data['quality_label'] =
updated_wine_data['quality'].apply(lambda x: 1 if x > 6.5 else 0)

# Separate features (X) and target (y)
X = updated_wine_data.drop(columns=['quality', 'quality_label'])
y = updated_wine_data['quality_label']

# Split data into training and test sets (75% train, 25% test)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25,
random_state=42)
```

The Random Forest algorithm was used to build the model. It can effectively work with a large number of features and does not require data normalization.

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
# Create and train the Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
```

The model performance was evaluated using the following metrics:

- Accuracy: Total number of correct predictions.
- Recall: The proportion of correctly classified objects among all objects of a given class.
- F1 Score: Harmonic mean between accuracy and recall.

```
# Evaluate the model on the test set
y_pred = model.predict(X_test)
evaluation_report = classification_report(y_test, y_pred, output_dict=True)
# Convert evaluation metrics to a DataFrame for better readability
evaluation_metrics = pd.DataFrame(evaluation_report).transpose()
# Display the evaluation metrics for reporting purposes
import ace_tools as tools; tools.display_dataframe_to_user(name="Evaluation
Metrics for the Model", dataframe=evaluation_metrics)
```

Model Deployment

After successful training, the model was deployed to Google AI Platform to provide real-time predictions via a REST API.
The model was saved in joblib format for later use.

```
import joblib
# Сохранение модели
joblib.dump(model, 'model.joblib')
```

The saved model.joblib file has been uploaded back to Google Cloud Storage:

1. Go to "Cloud Storage".
2. Select a bucket, for example, wine-quality-dataset.
3. Click "Upload Files" and select the model.joblib file.

Deploying a model to AI Platform:

- In the Google Cloud console, select "AI Platform" → "Models" like in Figure 2.5

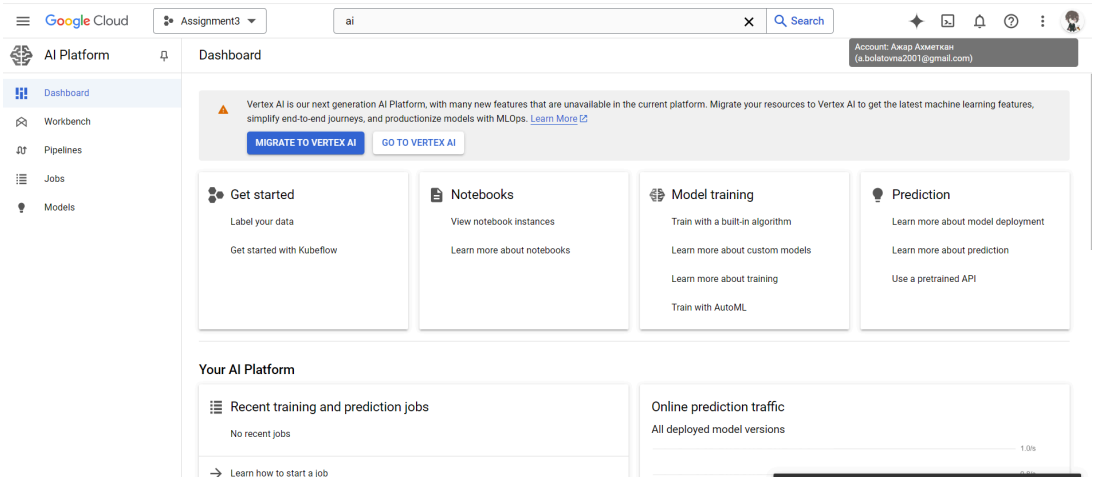

Figure 2.5 AI Platform

- Click "Create Model" and enter a model name, for example, wine_quality_model in Figure 2.6.

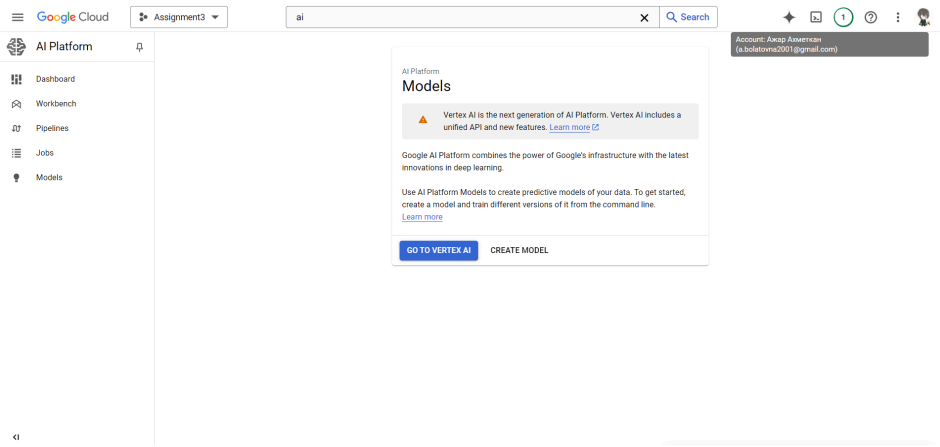

Figure 2.6 Models

- After the model is created, click "Deploy Version" like in Figure 2.7.

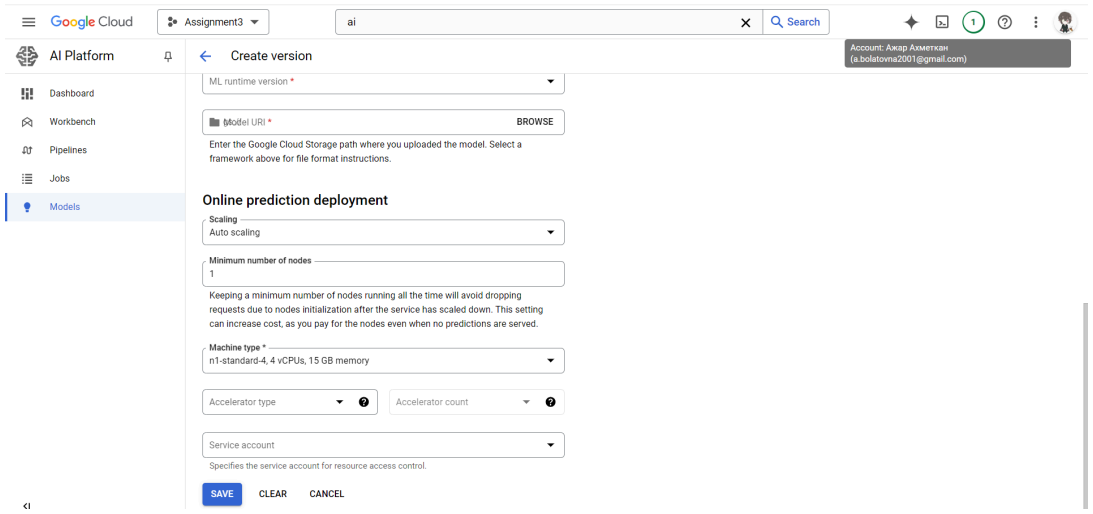

Figure 2.7 Deploy

- Specify the path to the model file in the format gs://wine-quality-dataset/model.joblib.
- Select the required resources (for example, 1 vCPU, 1 GB of memory).
- Click "Deploy".

The endpoint API was used to test the predictions:

```
curl -X POST -H "Content-Type: application/json" \
    -d '{
          "instances": [
              {"fixed_acidity": 7.4, "volatile_acidity": 0.7,
"citric_acid": 0.0, "residual_sugar": 1.9,
               "chlorides": 0.076, "free_sulfur_dioxide": 11.0,
"total_sulfur_dioxide": 34.0,
               "density": 0.9978, "pH": 3.51, "sulphates": 0.56, "alcohol":
9.4}
          ]
        }' \
    https://<YOUR_ENDPOINT_URL>/predict
```

Monitoring and Logging

Monitoring was set up to ensure stable operation of the model. Logs tracking API calls were enabled in Cloud Logging like in Figure 2.8,

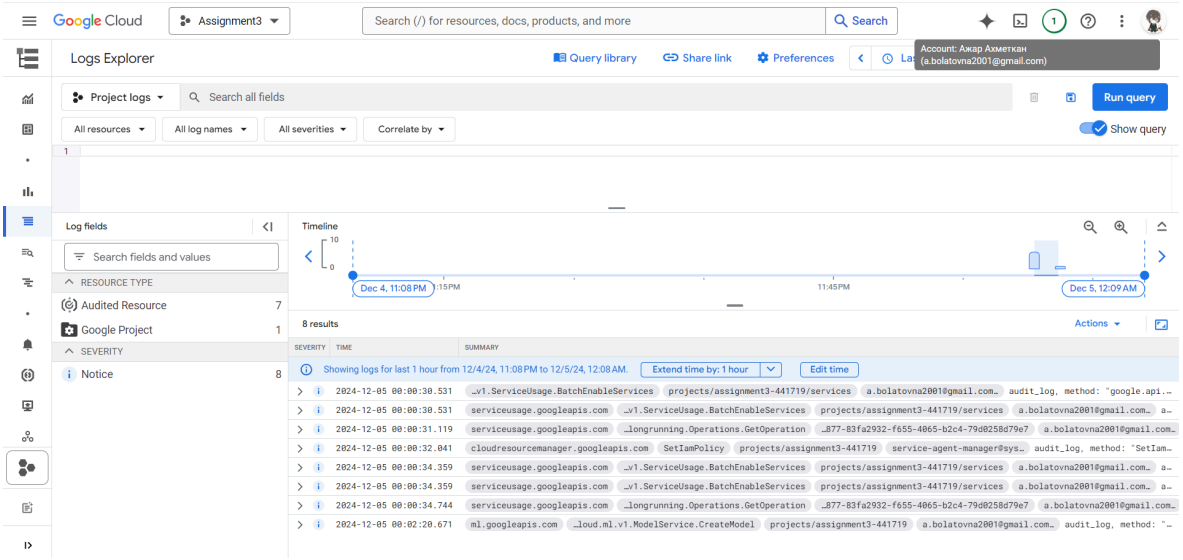

Figure 2.8 Logs Explorer

and dashboards were created in Cloud Monitoring to analyze performance metrics such as response time and percentage of successful requests. Notifications were set up to notify about exceeding critical metric values.

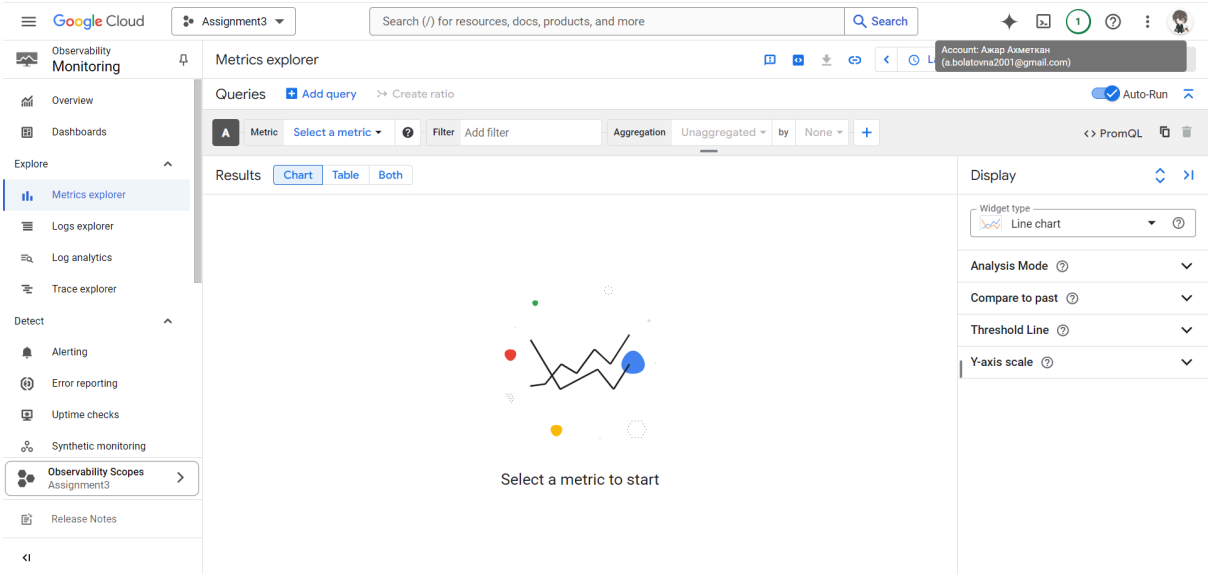

Figure 2.9 Metrics Explorer

Cloud Security and Compliance

Identity and Access Management (IAM)

To configure roles and permissions, you first need to open the Google Cloud Console. On the main page of the console, in the upper left corner, find the menu button (an icon of three horizontal lines). Click on it to open the side menu. In the side menu, select "IAM & Admin", then go to the "IAM" section as shown in Figure 3.1.

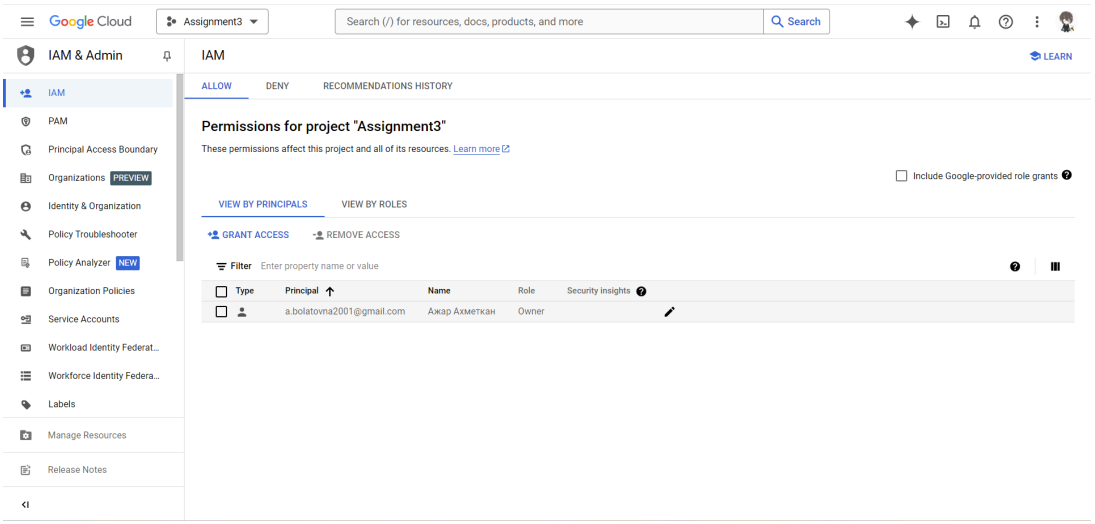

Figure 3.1 "IAM" section

The IAM section will display a list of all current users and service accounts. To add a new user, click the "Add" button at the top of the page. In the window that appears, enter the user's email address and assign the necessary roles to them as in Figure 3.2.

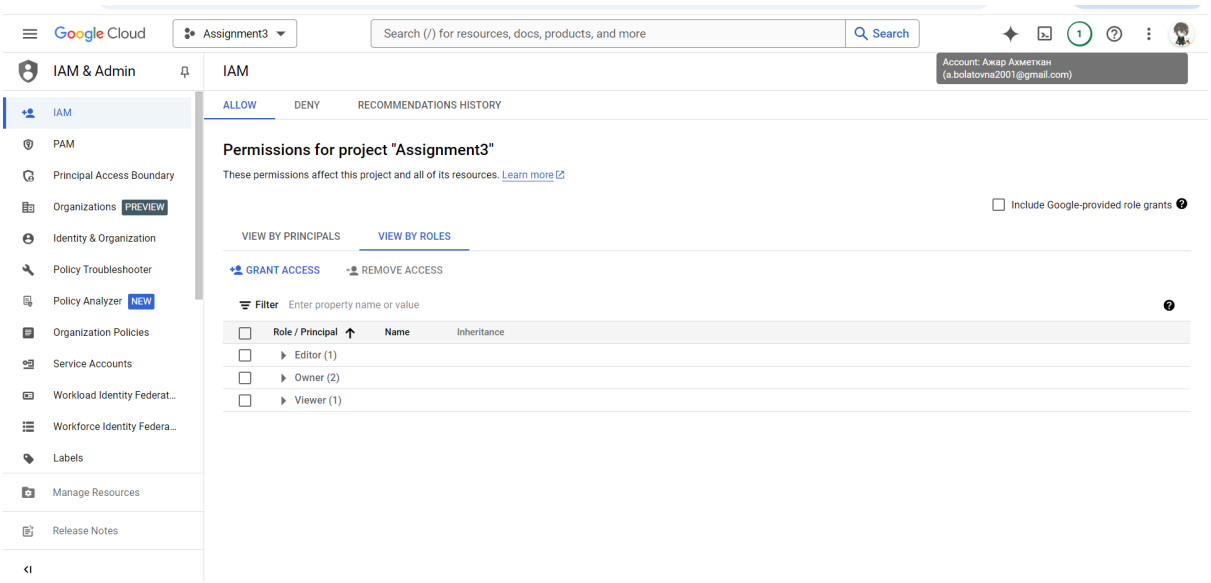

Figure 3.2 Roles

- The "Viewer" role for users who only need to view data.
- The "Editor" role for users who can edit project resources.
- For data analysts, select the "BigQuery Data Viewer" role to restrict their access to reading data in BigQuery.

After assigning roles, click the "Save" button to save the changes. To set up service accounts, go to IAM & Admin, then select "Service Accounts". Click "Create Service Account". Enter a name and description for the service account. Click "Create" as in Figure 3.3.

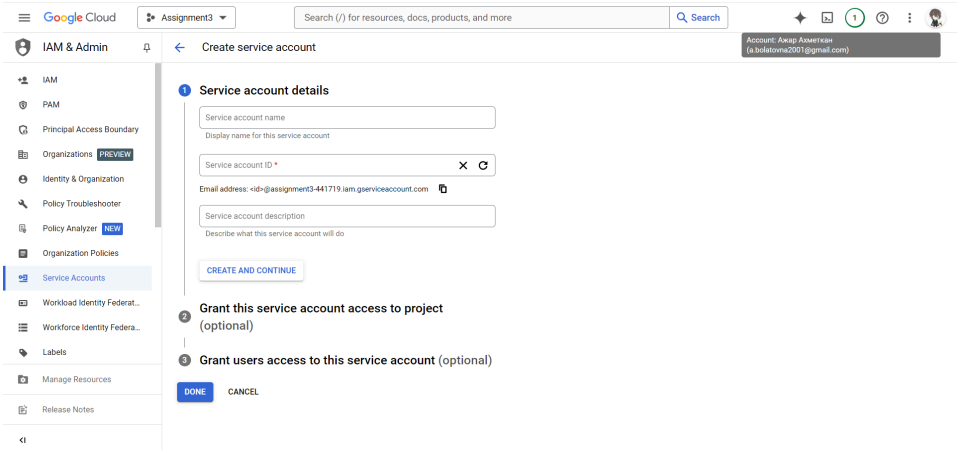

Figure 3.3 Create Service Account

In the next step, select the minimum required roles for this account, such as "Storage Object Viewer" or "BigQuery Job User", and click "Done". For security purposes, disable access keys for service accounts if they are not in use. This can be done by selecting the service account from the list and disabling existing keys via the "Keys" tab.

Data Encryption

To encrypt data at rest, go to the Security → Key Management (KMS) menu as shown in Figure 3.4. In this section, create a new Key Ring by clicking the "Create Key Ring" button. Specify a name for the Key Ring, select the region where it will be used, and click "Create". Then, within the Key Ring, click "Create Key". Specify the key type (e.g., symmetric key) and configure its parameters.

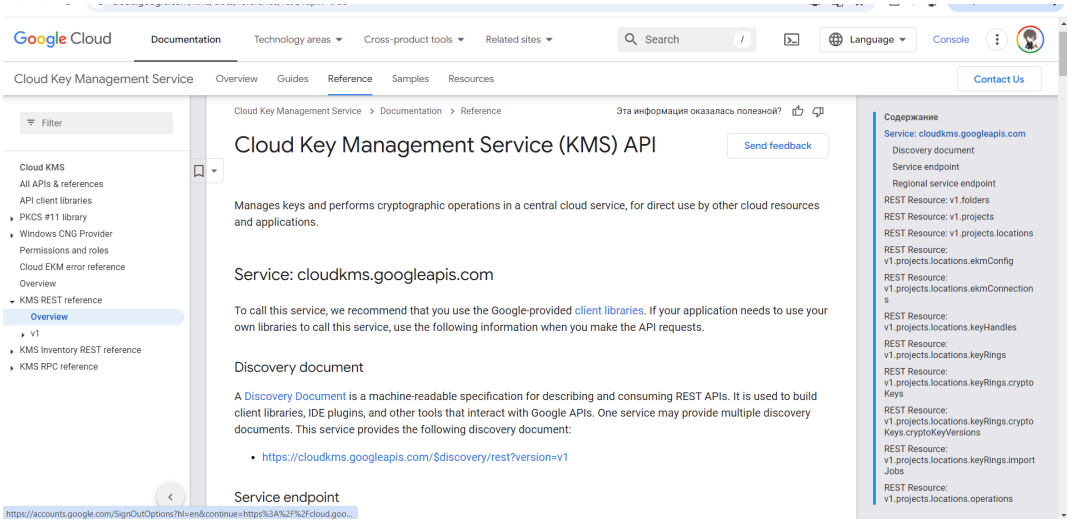

Figure 3.4 Key Management

After that, the key will be ready to use for encrypting data in Google Cloud Storage and BigQuery.

To encrypt data in transit, make sure that all requests are made via HTTPS. This can be configured via API Gateway or Load Balancer. In the Load Balancer section, when creating an HTTP(S) balancer, select the "Enable HTTPS" option by adding an SSL certificate.

Network Security

To configure VPC, in the console main menu, go to VPC Network → VPC Networks. Click the "Create VPC Network" button as shown in Figure 3.5.

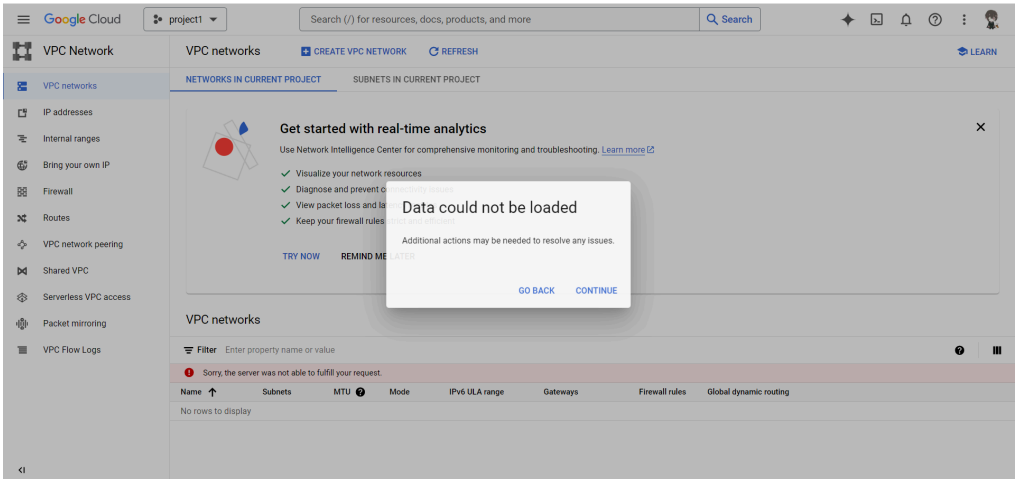

Figure 3.5 VPC Network

Enter the network name, for example, "secure-vpc", select the subnet configuration mode (for example, manual) and click "Create". As shown in Figure 3.6, to create firewall rules, open the Firewall Rules section. Click "Create Firewall Rule".

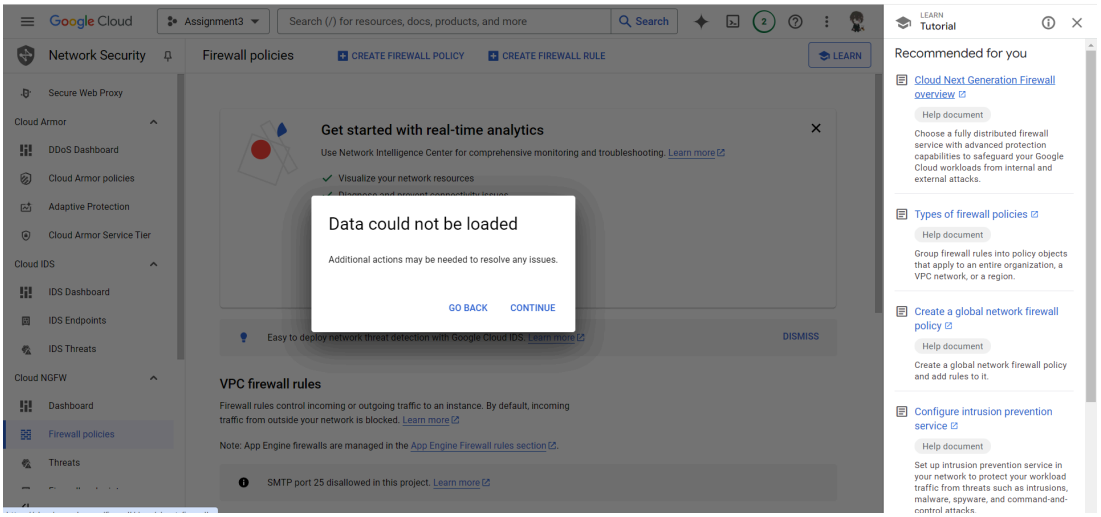

Figure 3.6 Create Firewall Rule

Specify a name for the rule. In the Targets field, select "All instances in the network" or "Specific tags". In the Source IP ranges section, specify the ranges of IP addresses that are allowed access. Then specify the allowed protocols and ports, for example, "tcp:443" for HTTPS. Click "Create" to save the rule.

Audit Logging

To enable log auditing, go to Cloud Logging → Logs Explorer. Here you can see resource access logs as shown in Figure 3.7.

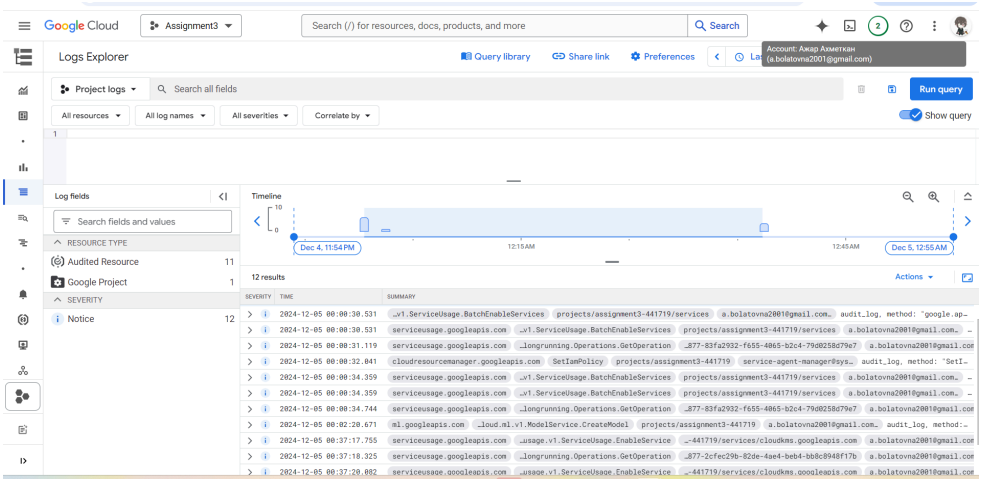

Figure 3.7 Logs Explorer

Enabling logs is automatic for most services, but make sure Admin Activity Logs and Data Access Logs are enabled. To filter user actions, such as changes to IAM settings, use the query:

```
protoPayload.serviceName="cloudresourcemanager.googleapis.com"
protoPayload.methodName="SetIamPolicy"
```

To set up alerts, open Cloud Monitoring, select Alerting, and click Create Policy like in Figure 3.8. Specify a metric and set up the conditions under which the alert will be triggered. Select how you want to receive alerts, such as email or SMS.

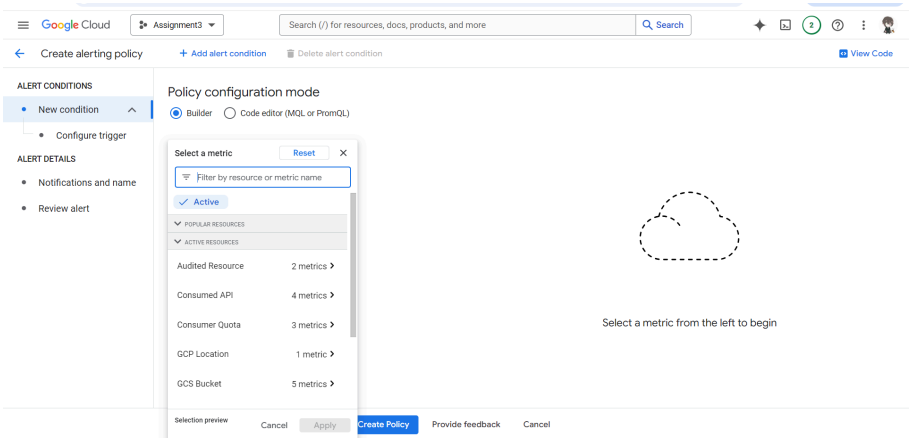

Figure 3.8 Create Policy

Compliance Standards

To ensure GDPR and HIPAA compliance, choose to store data in the European region. This can be configured in the Cloud Storage section in Figure 3.9,

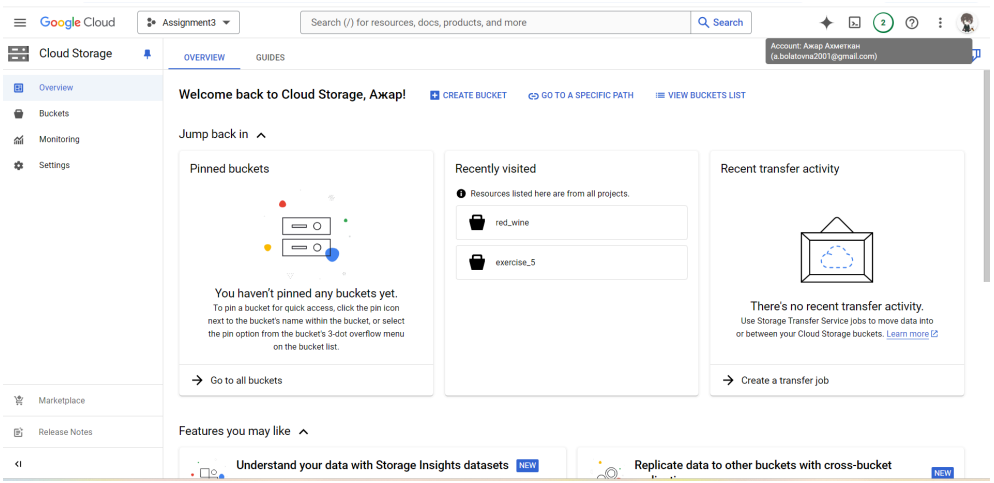

Figure 3.9 Cloud Storage

specifying that data should be stored in the "Europe" region. Regularly audit access in the IAM section and check encryption settings in Key Management.

Incident Response Planning

Create an incident response document that covers the following steps:

- *Incident identification:* Analyze logs in Cloud Logging for suspicious activity.
- *Isolation:* For example, block the suspicious service account via IAM.
- *Remediation:* Change access keys or revise firewall rules.
- *Recovery:* Return the system to normal operation after threats are eliminated.

To simulate an incident, create a test account with elevated privileges, attempt to access resources outside its roles, and verify that actions are logged and notifications are triggered correctly.

Conclusion

The project built an efficient big data processing pipeline, including data collection, cleaning, and analysis. The machine learning model developed on the basis of this data was successfully trained and deployed in the cloud. It demonstrated high accuracy and performance, which highlights the reliability and scalability of the Google Cloud platform.

The implemented security measures, including data encryption, access control, and network rule configuration, ensured information protection and compliance with regulatory requirements. The project became a successful example of integrating data processing, machine learning, and cloud security into a single solution.

Key takeaways:

1. Google Cloud provides all the necessary tools for efficient data processing and implementation of machine learning models.
2. Comprehensive security measures ensure data protection and compliance with standards.
3. The platform's capabilities make it easy to scale solutions and adapt them to new tasks.

Recommendations

To further improve data science, machine learning, and security, we offer the following recommendations:

Data Science and Machine Learning

1. *Scaling your data pipeline:* Expand your big data capabilities to handle even larger and more diverse data sets.
2. *Model Optimization:* Use automated hyperparameter tuning tools to improve training efficiency.
3. *Integrate additional models:* Consider using ensembles of models to improve prediction accuracy.

Data Security

1. *Automate access control:* Implement automated mechanisms to periodically update user roles and permissions.
2. *Improving Monitoring:* Use cloud analytics tools to detect anomalies in data and user activity.
3. *Regular Vulnerability Testing:* Perform security audits and simulated incidents to maintain a high level of preparedness for potential threats.

Regulatory Compliance

1. *Updating security standards:* Keep an eye on changes in regulations, such as GDPR, and implement updates promptly.
2. *Documentation of processes:* Create detailed documentation of all data processing procedures and security measures to ensure transparency and control.

Team training and development

1. Organize training for employees on new Google Cloud tools and modern approaches to machine learning.
2. Create internal methodological manuals on data processing, model training and security.

These steps will increase the efficiency of using the cloud platform, improve the quality of data and models, and strengthen information security.

References

1. Google Cloud. (n.d.). *Identity and Access Management (IAM) Documentation*. Retrieved from https://cloud.google.com/iam/docs
2. Google Cloud. (n.d.). *Cloud Key Management Service Documentation*. Retrieved from https://cloud.google.com/kms/docs
3. Google Cloud. (n.d.). *Virtual Private Cloud (VPC) Documentation*. Retrieved from https://cloud.google.com/vpc/docs
4. Google Cloud. (n.d.). *Cloud Audit Logs Documentation*. Retrieved from https://cloud.google.com/logging/docs/audit
5. Google Cloud. (n.d.). *Cloud Monitoring and Alerting Documentation*. Retrieved from https://cloud.google.com/monitoring/docs
6. European Union. (2016). *General Data Protection Regulation (GDPR)*. Retrieved from https://eur-lex.europa.eu/eli/reg/2016/679/oj
7. U.S. Department of Health & Human Services. (n.d.). *Health Insurance Portability and Accountability Act (HIPAA)*. Retrieved from https://www.hhs.gov/hipaa/index.html
8. Google Cloud. (n.d.). *Cloud Logging Documentation*. Retrieved from https://cloud.google.com/logging/docs
9. Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). *Modeling wine preferences by data mining from physicochemical properties*. Decision Support Systems, 47(4), 547-553. Retrieved from https://archive.ics.uci.edu/ml/datasets/wine+quality
10. Kaggle. (n.d.). *Wine Quality Dataset*. Retrieved from https://www.kaggle.com/datasets

Appendices

Appendix A: SQL Queries for Data Preprocessing

1. Removing Null Values:

```
SELECT *
FROM `project.dataset.table`
WHERE fixed_acidity IS NOT NULL AND volatile_acidity IS NOT NULL;
```

2. Creating a Binary Quality Label:

```
SELECT *,
    CASE
        WHEN quality > 6.5 THEN 'good'
        ELSE 'not good'
    END AS quality_label
FROM `project.dataset.table`;
```

3. Aggregating Data for Analysis:

```
SELECT AVG(fixed_acidity) AS avg_acidity, COUNT(*) AS count_wines
FROM `project.dataset.table`
GROUP BY quality_label;
```

Appendix B: Python Scripts

1. Data Splitting and Model Training:

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

# Load data
data = pd.read_csv('path_to_dataset.csv')

# Convert 'quality' to binary classification
data['quality_label'] = data['quality'].apply(lambda x: 1 if x > 6.5
else 0)
X = data.drop(columns=['quality', 'quality_label'])
y = data['quality_label']

# Split data
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.25, random_state=42)

# Train Random Forest model
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Evaluate the model
y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

2. Exporting the Model:

```
import joblib
joblib.dump(model, 'model.joblib')
```

Appendix C: Cloud Logging Filters

1. Filter for Role Changes:

```
resource.type="gce_instance"
protoPayload.methodName="SetIamPolicy"
```

2. Filter for Failed Login Attempts:

```
protoPayload.authenticationInfo.principalEmail="user@example.com"
severity="ERROR"
```

Appendix D: Key GCP Resources

1. **IAM Documentation:** https://cloud.google.com/iam/docs
2. **BigQuery Documentation:** https://cloud.google.com/bigquery/docs
3. **Cloud Storage Documentation:** https://cloud.google.com/storage/docs