

Google Cloud

A practical guide to AI with BigQuery



Table of contents

01

Introduction

1.1 The agentic AI era: From system of intelligence to system of action

05

1.2 The unified engine for data-to-AI: Accelerating enterprise value with BigQuery AI

06

02

A practitioner’s guide to the end-to-end AI workflow on BigQuery

2.1 Your central workspace: The AI-first, multilingual notebook

09

2.2 Accessing and preparing data without silos

12

2.3 A spectrum of model development options

14

2.4 From model to production with integrated MLOps

19

03

Set up: getting started with BigQuery AI

3.1 Google Cloud account

22

3.2 BigQuery Studio

23

3.3 Self-hosted JupyterLab

24

3.4 Self-hosted Visual Studio Code

25

04

Practical guide for BigQuery AI use cases

4.1 Use case 1: Modernizing retail demand forecasting

28

4.2 Use case 2: Analyzing healthcare documents

29

4.3 Use case 3: Decoding customer reviews

30

4.4 Use case 4: Automating contract analysis

31

4.5 Use case 5: Identifying customer segments

32

4.6 Use case 6: Building an image-based home search engine

33

4.7 Use case 7: Enabling conversational analytics

34

4.8 Use case 8: Preparing data with AI agents

35

01



Introduction

1.1 The agentic AI era: From system of intelligence to system of action	05
---	----

1.2 The unified engine for data-to-AI: Accelerating enterprise value with BigQuery AI	06
--	----



BigQuery offers comprehensive artificial intelligence (AI) and machine learning (ML) capabilities within the data platform to turn raw data into actionable insights. Referred to as BigQuery AI, these capabilities bring AI directly to data, rather than data to AI, simplifying the entire data lifecycle – from ingestion and preparation to analysis and machine learning – embedding Google’s state-of-the-art Gemini models and ML directly into BigQuery. By transforming the data platform into an autonomous data-to-AI platform, BigQuery AI empowers data teams to analyze complex datasets and run predictive workflows using standard SQL or Python, without needing complex data movements or learning new AI/ML skills. This guide helps you get started with AI workflows in BigQuery and offers a range of real-world use cases (accompanied by code) for you to explore.

The agentic AI era: From system of intelligence to system of action

Agentic AI is changing how we use data. Instead of simply storing information to look at later, data platforms are becoming active partners that can take real action. Modern data platforms are evolving from 'systems of intelligence' to 'systems of action' in three main ways:

01

From operating at human scale to agent scale

We are moving from humans asking a few questions per day to AI agents asking thousands of questions every minute. This means our systems must be much faster, more scalable, and highly affordable to keep up.

02

From reactive intelligence to proactive action

Instead of waiting for us to check a dashboard, AI spots trends and acts on them in real-time.

03

From data to semantic knowledge

Traditional data platforms use rigid schemas and stale metadata. AI agents use 'semantic search' to understand how different pieces of information relate to one another, much like a human would.

Modern data platforms like BigQuery offer agents and assistive AI capabilities that automate the tedious data-to-AI steps, allowing you to focus your time and expertise on what matters most: analysis, interpretation, and solving core business problems.

The unified engine for data-to-AI: Accelerating enterprise value with BigQuery AI

For years, the standard approach to AI and machine learning required a costly, slow, and complex cycle: exporting massive datasets from your enterprise data platforms – such as data warehouses, data lakes, and lakehouses – transforming them, and then loading them into specialized ML environments.

BigQuery changes everything. By unifying predictive ML, generative AI, semantic search, and autonomous agents directly within your data platform through BigQuery AI, BigQuery eliminates data movement and transformation, maintains strict security and governance, and speeds up model deployment from months to minutes. Organizations can now run powerful models – such as Gemini – right where their data resides to unlock real-time insights without operational friction using familiar tools and languages like SQL and Python.

BigQuery AI capabilities

To help you visualize how these capabilities translate to business impact, here is a complete breakdown of BigQuery’s five core AI capabilities:

AI capability	How it works in BigQuery	Business ROI and value
Predictive machine learning (BigQuery ML)	Train, evaluate, and deploy models (for example, time series, anomaly detection, recommendations, clustering) using SQL, Python, REST APIs, or integrated Colab Enterprise notebooks.	Democratizes ML: Enables existing SQL practitioners to build models instantly, drastically accelerating forecasting and anomaly detection on your core datasets.
Generative AI and AI functions	Access Gemini and partner Large Language Models (LLMs) directly in your queries. Use general-purpose gen AI functions for full prompt/model control, or managed AI functions for automated tasks (for example, filtering, sentiment tracking, classification, and document processing).	Massive scale and efficiency: Automates text generation, translation, document processing, and unstructured data enrichment directly within your data platform, eliminating costly external API pipelines and optimizing token usage.
Vector and multimodal search	Generate embeddings and execute VECTOR_SEARCH or AI.SEARCH over multimodal data with Approximate Nearest Neighbor (ANN) indexing.	Enables next-gen apps: Powers real-time, highly relevant recommendations, semantic search engines, and Retrieval-Augmented Generation (RAG) applications directly on top of your live data.
Conversational analytics	Natural language interaction to discover, clean, query, and visualize data, alongside SQL/Python code completion and explanation.	Boosts developer productivity: Drastically lowers the technical barrier, helping analysts generate immediate insights and transform raw data into beautiful visual charts using plain language.
Data agents	Built-in agentic workflows (such as Data Science, Data Engineering, and Conversational Analytics agents) that execute tasks on your behalf.	Automates end-to-end data pipeline creation, exploratory data analysis, and conversational analysis using natural language prompts.

02



A practitioner's guide to the end-to-end AI workflow on BigQuery

2.1. Your central workspace: The AI-first, multilingual notebook	09
2.2. Accessing and preparing data without silos	12
2.3. A spectrum of model development options	14
2.4. From model to production with integrated MLOps	19

As a business analyst or a data scientist, your goal is to get from a business question to a robust, actionable insight as efficiently as possible. In this section, we'll explore a practical walkthrough of Google Cloud and third-party tools that support this modern workflow. You'll learn how to move from set up to data access, model building, and production – all within a unified ecosystem and using popular tools like IDEs, notebooks, APIs, and SDKs to keep you in your analytical flow and quickly go from data to predictive insights



Your central workspace: The AI-first, multilingual notebook

Notebooks are the preferred development environment for AI projects. However, a fragmented environment forces you to constantly switch tools, which switches context and breaks flow. For example, a typical workflow involves writing SQL in one tool for exploratory analysis or data transformation, exporting results, loading them into a local Python notebook, and then using a different tool for visualization.

Google Cloud offers a flexible suite of notebook solutions tailored to enterprise business analysts and data scientists, accommodating various workflows and governance requirements. For a fully managed and collaborative notebook experience within BigQuery, [Colab Enterprise](#) notebooks in BigQuery provides a secure, familiar interface with deep integrations with Gemini Enterprise Agent Platform and BigQuery. For greater customization within a managed framework, Gemini Enterprise

Agent Platform Workbench offers powerful JupyterLab instances that serve as a direct entry point to Serverless Spark, BigQuery, and [Gemini Enterprise Agent Platform](#). Teams preferring a full IDE experience can use **Cloud Workstations** equipped with Cloud Code extension to provision secure, on-demand Visual Studio Code environments in the cloud. [The Google Cloud Data Agent Kit](#) is an open-source development package that allows data practitioners to connect to, orchestrate, and manage Google Cloud data workloads

directly from their local coding environments. Finally, for those who prefer their own self-hosted setups, Google Cloud bridges the gap with tools like the **BigQuery JupyterLab** and **Visual Studio Code** extensions, enabling seamless data exploration and machine learning directly from a user's preferred IDE. The remainder of this section focuses on [Colab Enterprise notebooks in BigQuery](#).

Colab Enterprise notebooks

Colab Enterprise notebooks in BigQuery let you perform end-to-end AI workflows within a single, integrated interface. Unlike standard SQL editors, notebooks let you combine SQL queries with Python code, rich text, and visualizations to let your data tell a comprehensive story. Notebooks are ideal for the following use cases:

End-to-end AI workflows: Build, evaluate, and deploy a BigQuery ML model within a single notebook interface.

Data exploration: Clean and analyze large datasets using BigQuery DataFrames.

Collaborative research: Share notebooks with colleagues using IAM and track version history.

Notebooks are code assets in BigQuery Studio (BigQuery's workspace), alongside saved queries. Benefits of notebooks include:

Seamlessly blend languages: This is where the friction of context switching disappears. In this polyglot environment, you can query data using SQL in one cell and have the output instantly available as a Python DataFrame in the next. You can also use Spark for large-scale transformations without ever leaving the notebook interface.

An AI-assisted workflow: Colab Enterprise notebooks are designed with an integrated Data Science Agent to act as an intelligent partner. You can start with a high-level goal in plain English, and the agent can generate a multi-step plan and the corresponding code: you remain in control, reviewing and editing each step.

Familiar editor features: Use SQL auto-completion, similar to the BigQuery SQL editor.

Integrated visualizations: Use interactive DataFrame visualizations, or libraries like matplotlib and seaborn, to visualize data directly in your workflow.

SQL-Python interoperability: Execute SQL in cells that reference Python variables.



A practical example

Your prompt:

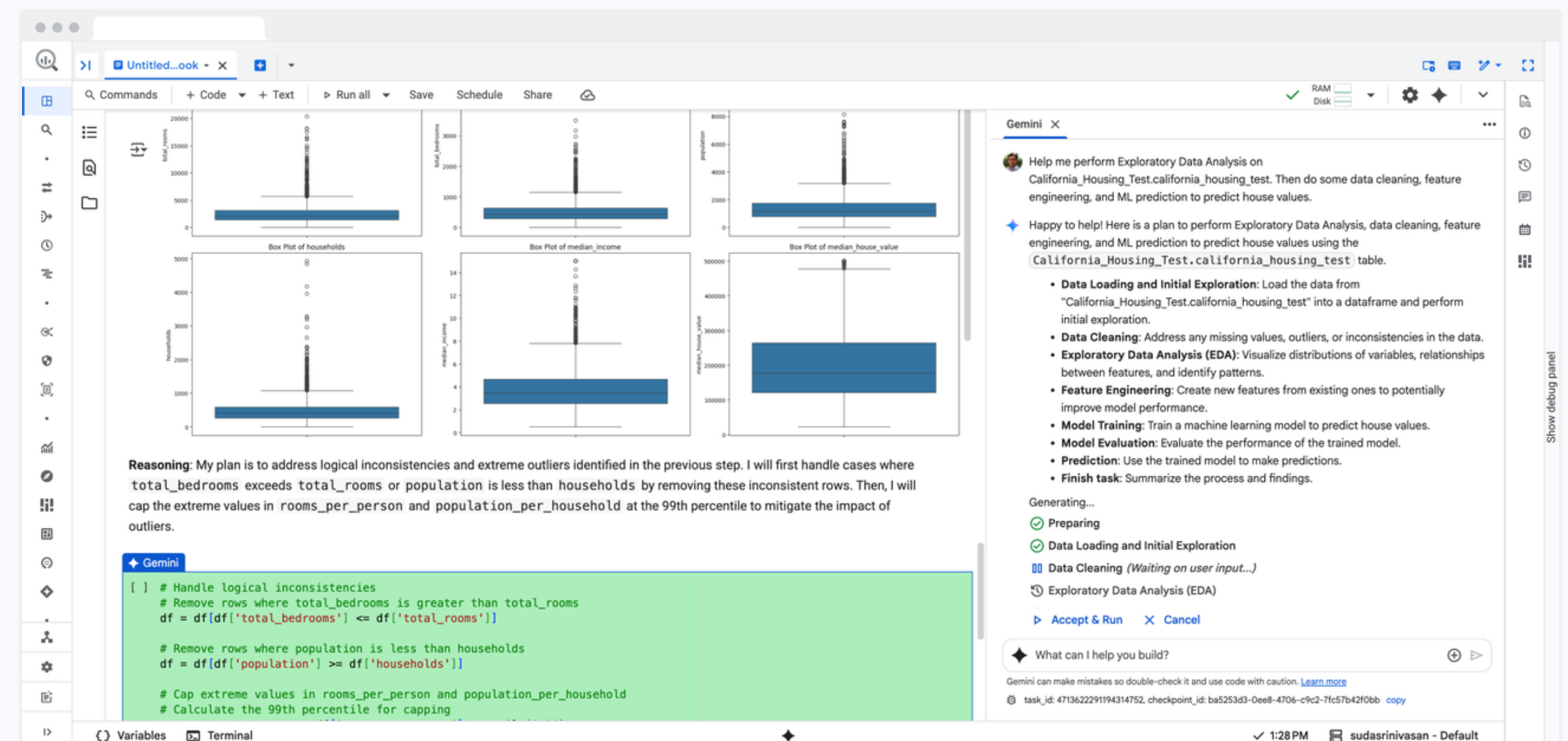
Train a model to predict 'income_bracket' from table: census_adult_income dataset.”

AI-generated plan:

01. Load the census_adult_income table from BigQuery
02. Clean the data by removing rows with null values
03. Train a logistic regression model using BigQuery ML to predict income_bracket
04. Evaluate the model's accuracy

The agent then generates the SQL and Python code for each step, which you can execute, modify, and build upon. AI-powered code completion, error explanation and correction, and visualization further simplify the experience.

Colab Enterprise notebooks offer a unified, AI-powered development environment for analysis and rapid experimentation.



Accessing and preparing data without silos

Your environment is now set up; next you need data. This step is where friction often arises, from handling different data types to rewriting code when datasets become too large for a single machine.

A unified, multimodal data foundation

Typically, only a small part of your data is structured, with insights often locked in images, documents, and audio files. BigQuery allows you to access all this data in a borderless way. Using Object Tables and the ObjectRef data type, you can treat unstructured data in Google Cloud Storage as if it were a standard SQL table. This means you can use the power and familiarity of SQL to analyze images or process documents directly. For example, you can call an AI function to extract text from a PDF or analyze an image within your SQL query. Beyond breaking down the data silos, all this data, whether structured or unstructured, is secured under a single, unified governance model, simplifying compliance

AI-assisted data preparation

Data scientists usually spend a lot of time on manual, or repetitive data cleaning and transformation tasks. BigQuery accelerates these processes with AI-assisted data preparation. The built-in Gemini-powered editor transforms data preparation from a manual coding exercise into an interactive, AI-guided workflow that allows you to focus on the logic of your transformations rather than the syntax. Additionally, as you work with your data, Gemini provides context-aware suggestions for tasks – like applying transformations, standardizing data, and automating schema mapping.

The new **Data Engineering Agent** in BigQuery goes beyond assistive AI and provides autonomous agentic capabilities. The data engineering agent takes instructions in plain English, like "Create a pipeline to load

data from the 'customer_orders' bucket, standardize the date formats, remove duplicate entries based on order ID, and load it into a BigQuery table named 'clean_orders.'" and generates the required SQL code, builds the pipeline, and even creates basic unit tests. The agent can handle a wide range of data transformations and data engineering tasks, including ingestion, data quality, incorporating custom business logic in pipelines, and data modeling.



Want to learn more about working with multimodal data in BigQuery? Explore this notebook with code samples:
goo.gle/bq-multimodal-sample

Data enrichment with Google and third-party datasets

Using the integrated Marketplace in BigQuery you can discover and use over 3,500 commercial and public datasets, joining them with your own tables without any data movement or ETL processes. This lets you leverage unique public datasets from Google, including the Earth Engine Data Catalog for planetary-scale environmental data, Google Places Insights for location-based intelligence, and Google Trends to analyze search interest over time. Combining your proprietary business data with global-scale external intelligence lets you dig deeper for insights.

Flexible data processing with multiple engines

Different tasks require different tools. BigQuery's architecture allows you to bring your preferred processing engine –whether it's BigQuery's SQL engine or an open-source framework like Apache Spark – directly to a single, unified copy of your data. This avoids the need to maintain separate data copies for different systems.



“Public Value Technologies receives data feeds from our media partners for our data mesh solution and AI applications. BigQuery data preparation allows our data analysts and scientists to rapidly integrate the data feeds that standardize and preprocess the data in a low code way.”

Korbinian Schwinger, Team Lead Data Engineer, Public Value Technologies

A spectrum of model development options

BigQuery provides a broad set of capabilities designed to meet you where you are, whether you prefer the speed of SQL, the power of pre-trained foundation models, or full control over custom model development.

Bringing AI to your data with BigQuery ML

A significant amount of time on any AI project is spent engineering pipelines simply to move data to a separate environment for model training. This forces users to deal with separate infrastructure and security policies, breaking flow and causing friction. BigQuery ML (BQML) eliminates this by allowing you to train, evaluate, and deploy machine learning models directly within the data platform using familiar SQL commands. This is a practical approach for quickly building models like linear regression, k-means clustering, classification, or time-series forecasts without moving data. It also democratizes model building, enabling data analysts who are already SQL experts to develop powerful predictive models.



```
CREATE OR REPLACE MODEL 'my_dataset.income_prediction_model'

OPTIONS(model_type='LOGISTIC_REG') AS

SELECT

    * EXCEPT(income_bracket),

    income_bracket as label

FROM `bigquery-public-data.ml_datasets.census_adult_income`;
```

Aha! Moment: train an ML model with SQL

“With BigQuery ML, we've accelerated our model deployment to production 10x faster, enabling our sales teams to better serve our Pro customers with prioritized leads and deeper insights.”

Mani Pantangi, Senior Manager, Data Science, The Home Depot





BQML offers a wide range of commonly used statistical and predictive machine learning models that are built into BigQuery for ease of use, including contribution analysis, linear regression, classification, k-means clustering, matrix factorization, time series forecasting, and more. These built-in models greatly simplify the most common workflows for data scientists. As an added benefit, you can perform a dry run before creating the model to get an estimate of how much data the model will process to run.

Leveraging pre-trained foundation models

Rather than building every model from scratch, you can command powerful, pre-trained foundation models.

Generative AI: BigQuery lets you use SQL functions that call on LLMs like Gemini to analyze or enrich your data. For example, you can use the AI.GENERATE function which, under the hood, uses Gemini to analyze any combination of structured and unstructured data to generate text or structured output according to a custom schema that you specify.

Specialized foundation models: For specific tasks, you can use state-of-the-art models directly. For example, TimesFM and TabFM, models from Google Research that have been pre-trained on over 100 billion data points, are integrated into BQML. This lets you generate highly accurate time-series forecasts and regressions/classifications with a simple SQL query, bypassing weeks of custom model development.

Optimized mode for AI functions in BigQuery: BigQuery's optimized AI functions, such as AI.IF and AI.CLASSIFY, reduce your token consumption by 230x compared to standard row-by-row generative AI processing powered by distilled models that are trained on the fly for your specific query, a breakthrough innovation that provides scalable AI query execution while reducing LLM token costs.

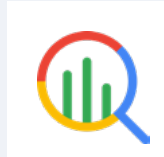
Want to learn more about working with AI functions in BigQuery? This notebook has code samples to help you get familiar with our full range of AI functions: goo.gle/bq-ai-functions-sample.

Custom model development with Gemini Enterprise Agent Platform: For projects that need more control, Gemini Enterprise Agent Platform provides an end-to-end ML platform. Here, you can build, train, and tune custom models using PyTorch, TensorFlow, or any other ML library. The power lies in the seamless integration: you can do feature engineering in BigQuery, train a custom model in Gemini Enterprise Agent Platform, and then use that model back into BigQuery for inference via SQL.

Want to learn more about working with Gemini Enterprise Agent Platform and BigQuery to build and train a custom ML model? This tutorial helps you learn more: <https://codelabs.developers.google.com/vertex-workbench-intro#0>

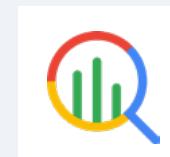


When to use which tool? A quick guide



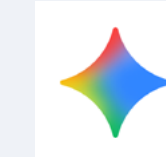
Use BigQuery ML

When you want to rapidly build and iterate common model types (such as regression, classification, clustering, or forecasting) directly on your data in BigQuery using SQL. This is ideal for fast results and teams with strong SQL skills and basic ML understanding.



Use BigQuery AI functions

When you want SQL functions for generative AI tasks such as text generation, text or unstructured data analysis, and translation. These functions access pre-trained Gemini and partner LLM models available from Gemini Enterprise Agent Platform, Google Cloud AI APIs, or built-in BigQuery models to perform these tasks.



Use Gemini Enterprise Agent Platform

When you need to build a custom model with a specific architecture, require advanced MLOps capabilities like experiment tracking, or need to use Python frameworks like TensorFlow or PyTorch for deep customization and control.

Vector embeddings and vector search:

Beyond generating insights from individual data points, a common AI task is to find similar items within a large dataset. BigQuery lets you generate and use vector embeddings to perform vector search, enabling semantic understanding and similarity-based retrieval of multimodal data.

The workflow to generate embeddings is straightforward: BigQuery's AI.EMBED is a built-in SQL function that converts text or images into embeddings directly within your data platform. Best of all, you

can either run it on-demand with remote Gemini Enterprise Agent Platform models or configure fully managed, asynchronous pipelines where BigQuery automatically generates and synchronizes these vector columns in the background as your data changes. You can create vector indexes using algorithms like Inverted File (IVF) indexing or the Scalable approximate Nearest Neighbor (ScaNN) algorithm from DeepMind to accelerate similarity searches. This allows you to build sophisticated semantic search, recommendation, or segmentation systems

without needing to manage a separate, specialized vector database. BigQuery is especially useful for these systems because it is all managed serverlessly, regardless of your batch or interactive needs.

To solve inconsistent product matching, [Mercado Libre](#) used BigQuery vector search on product embeddings. It acted as a high-speed filter, rapidly identifying an initial set of similar items from their vast catalog. This crucial first step enabled other AI models to then perform a more detailed analysis on a pre-filtered list

From model to production with integrated MLOps

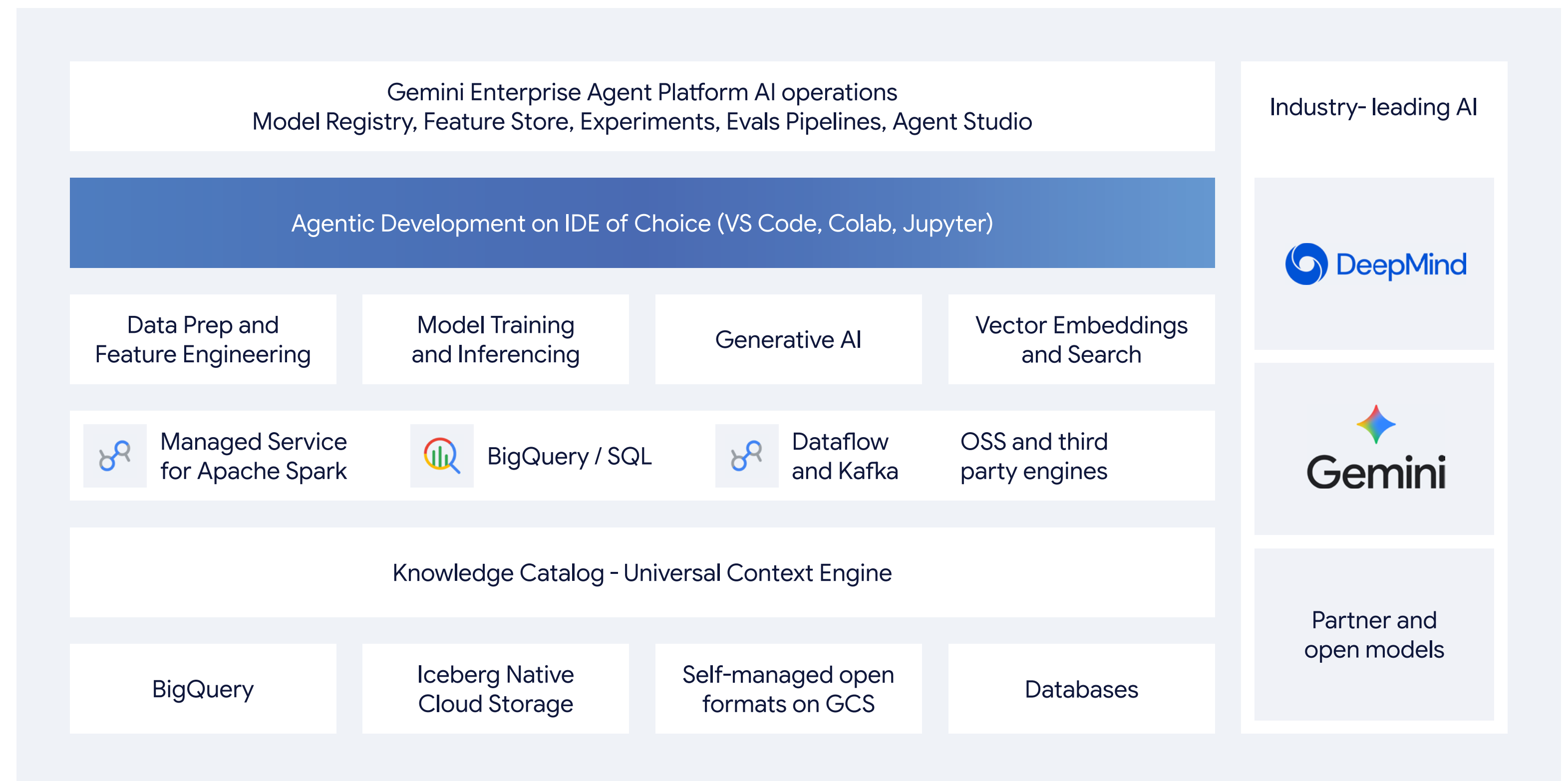
A model only provides value when it's operationalized. The 'last mile' MLOps challenge of moving a model from a notebook to a scalable, monitored production environment is addressed through a tight integration between BigQuery and Gemini Enterprise Agent Platform.

Google Cloud offers a suite of capabilities to simplify data science workflows

Automated model building and management:

For tabular data, Gemini Enterprise Agent Platform AutoML can automatically handle feature engineering, model selection, and hyperparameter tuning to produce an optimized model.

Model management and deployment: Models trained in BigQuery ML can be registered in the Gemini Enterprise Agent Platform Model Registry. From there, they can then be versioned, evaluated, and deployed for batch or online predictions with just a few clicks, completing the end-to-end lifecycle from a single, integrated platform.



03

Set up: Getting started with BigQuery AI



3.1 Google Cloud account	22
3.2 BigQuery Studio	23
3.3 Self-hosted JupyterLab	24
3.4 Self-hosted Visual Studio Code	25



So far, we've talked about key Google Cloud capabilities to streamline and simplify AI workflows. In this section and the next, you'll gain practical insights into these processes. We'll first explore what tools you need to start exploring AI use cases and get hands-on experience, and then consider some common BigQuery use cases for business analysts and data scientists.

Google Cloud account

To start using Google Cloud services, you first need to have an account with Google Cloud. If you are new to Google Cloud and don't have an existing account, you can sign up for a [90 day free trial](#) that includes \$300 in free cloud billing credits. You can use these credits toward one or a combination of products. Additionally, Google Cloud accounts also provide a [free tier](#) to several Google Cloud products, including BigQuery and Cloud Storage, providing limited access at no cost. Once your Google Cloud account is set up, you will need to decide what notebook platform fits your specific needs.

As mentioned in Chapter 2, you have several options to choose from, including BigQuery Studio, Gemini Enterprise Agent Platform Workbench, Cloud Workstations, and self-hosted JupyterLab or Visual Studio Code. In the following sections, you'll learn how to get started with each of these options.



BigQuery Studio

To begin using BigQuery Studio:

01

Navigate to the [BigQuery page](#) in the Google Cloud console.

02

If prompted, enable any required APIs, such as the Dataform API, to activate all of its features.

03

Once the interface loads, in the tab bar of the editor pane, select the down arrow next to the plus sign. Then select Notebook » Empty Notebook from the floating menu.

04

If it's your first time creating a code asset, you may also be prompted to choose a default storage region.

05

A new tab will then open, presenting you with a blank, untitled Colab notebook – ready for you to start writing Python and SQL code.

06

You will also have the option of starting with a template or uploading a notebook from your computer.

For more information, please visit the [Create notebooks page](#) in the documentation.



Self-hosted JupyterLab

The BigQuery Jupyter extension is a JupyterLab extension that brings powerful Google Cloud features directly into a user's local JupyterLab environment. This allows business analysts and data scientists to treat BigQuery or Spark, on Managed Service for Apache Spark, as a seamless backend for their notebooks. Key features include the ability to run code on BigQuery and Spark (serverless or clusters) without any local Spark installation, submit PySpark jobs directly from the notebook, browse and edit objects on Cloud Storage, and

browse and preview BigQuery datasets – all from within the familiar JupyterLab interface. This dramatically simplifies the workflow for developing and deploying AI pipelines on managed infrastructure. To install the BigQuery Jupyter extension and start your first workflow, visit the [Use the BigQuery Jupyterlab plugin](#) page in the documentation.

Self-hosted Visual Studio Code

The Google Cloud Code extension and Data Agent Kit transforms Visual Studio Code into a powerful, integrated development environment for data professionals on Google Cloud. This allows for direct interaction with BigQuery, enabling users to seamlessly browse datasets, inspect schemas, and preview table data without leaving the IDE. For Spark, the Managed

Service for Apache Spark extension allows users to manage the entire cluster and job lifecycles. You can execute notebooks on Managed Service for Apache Spark clusters and serverless, submit jobs to both traditional Managed Service for Apache Spark clusters and serverless, and create and manage reusable serverless Spark runtime templates to ensure your

notebooks execute with consistent Spark configurations. To install the Google Cloud Code extension for Visual Studio Code and use it, visit the [Install the Cloud Code for VS Code extension page](#) in the documentation. For more information on the Data Agent Kit, visit the [Google Cloud Data Agent Kit extension](#) documentation.



04

Practical guide for BigQuery AI use cases



4.1 Use case 1: Modernizing retail demand forecasting	28
4.2 Use case 2: Analyzing healthcare documents	29
4.3 Use case 3: Decoding customer reviews	30
4.4 Use case 4: Automating contract analysis	31
4.5 Use case 5: Identifying customer segments	32
4.6 Use case 6: Building an image-based home search engine	33
4.7 Use case 7: Enabling conversational analytics	34
4.8 Use case 8: Preparing data with AI agents	35

Now that your AI environment is set up and ready to go, this section looks at some practical, scenario-based use cases that you can explore using BigQuery. Each use case starts with a common business challenge and walks you through how to solve it using Google Cloud tools. The examples below explore different statistical and machine learning workflows that business analysts and data scientists often use.



Modernizing retail demand forecasting

The business challenge

In retail, accurate demand forecasting is critical for managing inventory. Overestimating demand leads to high storage costs and wasted products, while underestimating demand results in stockouts and lost sales. The core challenge lies in generating forecasts that are not only accurate, but also timely and highly granular.

A business rarely needs to predict only its total sales. To make effective operational decisions, it must forecast demand for every single item in each specific store or region. Current approaches are complex and resource-intensive. They require AI teams to build, manage, and retrain thousands of individual models – often one model for each unique product-location combination. This process can be slow, computationally expensive, and difficult to maintain, making it hard for businesses to react quickly to changing market conditions.

This use case is based on the public [Iowa Liquor Sales dataset](#). It provides a practical example, containing daily sales records for various products across the state. While it focuses on liquor sales, the principles and techniques are directly transferable to any retail scenario. The fundamental challenge is the same – whether you are forecasting demand for a specific t-shirt size in a particular store, a smartphone model in a city, or a brand of cereal in a specific supermarket. The goal is to find a scalable process that can handle this complexity efficiently, directly where the data resides.

The BigQuery AI approach

Implement and compare two powerful time series models using [BigQuery Machine Learning](#), eliminating the need to move massive datasets to a separate environment for training. First, use [ARIMA](#), a well-established statistical model, which requires us to train on historical data before generating a forecast. Then, use BigQuery's [AI.FORECAST](#) function, which uses [TimesFM](#), a foundation model from Google Research, to produce zero-shot forecasts without a separate training step. Compare these methods to evaluate the tradeoffs between the two approaches and pick one that best meets your needs.

High-level flow

- 01. Prepare the data:** Aggregate raw sales data into two distinct tables: one for a single time series and another for a multiple, granular time series.
- 02. Forecast a single time series:** Forecast total sales over time using both the ARIMA and TimesFM models.
- 03. Visualize single series results:** Plot forecasts from both models to assess their performance and differences.
- 04. Forecast multiple time series:** Scale the analysis to forecast sales for each county and item_name combination, handling thousands of series at once.
- 05. Visualize multiple series results:** Finally, visually compare forecasts for a specific county-item pair to compare the model's performance at a granular level.

Analyzing healthcare documents

The business challenge

In clinical research, a massive volume of critical, life-saving data remains locked away in unstructured formats, such as dense PDFs containing clinical trial protocols, complex inclusion/exclusion criteria, medical literature, and hand-written clinical patient notes. Historically, extracting these insights and matching a patient's natural language symptoms to rigid clinical trial criteria required slow, manual reviews, disjointed natural language processing pipelines, and complex data movement across separate vector databases and analytics platforms. This operationally fragmented approach significantly delays trial matching, increases administrative costs, and limits the scale at which healthcare and life sciences organizations can successfully identify eligible candidates for clinical trials.

The BigQuery AI approach

BigQuery serves as a unified, serverless AI and data platform that bridges unstructured text and structured insights securely within BigQuery's boundaries, eliminating data movement and infrastructure management. By utilizing Document AI and Gemini models directly within BigQuery via simple SQL-based commands, the platform parses raw PDF documents, categorizes clinical notes using AI functions like `AI.CLASSIFY`, and flags trial exclusions via `AI.IF`. Furthermore, BigQuery's native embedding generation (`AI.EMBED`) autonomously vectorizes unstructured text to match patient symptoms to medical ontologies, while its built-in graph capabilities (BigQuery Graph) allow researchers to seamlessly traverse and query complex, multi-hop relationships (such as mapping: Patient » Symptom » Disorder » Trial) at petabyte scale.

High-level flow

- 01. Access unstructured clinical trial files:** Upload unstructured clinical files, such as PDFs containing clinical trial protocols or Text files containing patient notes, to a Google Cloud Storage bucket and access them via BigQuery Object Table.
- 02. Parse unstructured data and organize the information:** Extract structured and unstructured metadata and data to BigQuery tables by parsing unstructured data using Document AI and BigQuery AI functions like `AI.PARSE_DOCUMENT`.
- 03. Vector embeddings and search:** Automatically vectorize the unstructured clinical text into high-dimensional vector spaces for semantic searching using BigQuery AI functions like `AI.EMBED`.
- 04. Map and visualize data using graphs:** Create property graphs that map and visualize relationships between patients, symptoms, and active trials using BigQuery Graph.
- 05. Get actionable insights:** Finally, write automated patient-friendly eligibility summaries using multimodal LLMs (like Gemini), and execute multi-hop queries – such as finding active Phase 3 recruiting trials for specific drug matches – using Graph Query Language (GQL), completing the entire end-to-end flow inside BigQuery.

Decoding customer reviews

The business challenge

One of the most urgent and ambiguous challenges a business can face is an unexpected dip in sales for a new product. The initial sales dashboards clearly show what is happening – revenue is falling short of forecasts – but they offer no insight into why. The potential causes are numerous and complex. Is it a pricing issue? A competitor's surprise launch? An ineffective marketing campaign? Or is there a fundamental flaw in the product itself?

In these situations, standard data analysis is often not enough. The real story may be hidden in plain sight or buried deep within heaps of disconnected data. This is a classic scenario where a data scientist must think like a detective. Their role isn't just to report on metrics but to sift through open-ended and often unstructured data, frame specific hypotheses, and then test them to uncover the truth. Critically, time is of the essence; while lagging sales puts ever more pressure on the business, the outcome of the investigation can directly influence critical product and go-to-market decisions. The challenge is to move from a vague business problem to a precise, data-driven explanation that can guide immediate action, and to do so quickly.

The BigQuery AI approach

BigQuery converts unstructured customer feedback into a powerful, structured signal for predictive modeling. The core of the approach is to use generative AI as a sophisticated feature engineering engine. Instead of just performing high-level sentiment analysis, this approach uses AI directly within BigQuery to parse thousands of text reviews and extract structured, nuanced information – such as sentiment scores for specific product features like 'battery life' or 'user interface.'

This crucial step transforms raw, unstructured text into a queryable, structured format. This new, AI-generated data can then be seamlessly joined with traditional structured business data, such as sales transactions and customer profiles. With this enriched, unified dataset, data scientists can apply familiar and powerful machine learning techniques. By using an out-of-the-box function like `AI.KEY_DRIVERS` or training an XGBoost model, for example, data scientists can move beyond correlation to understand causation, precisely identifying which specific product attributes mentioned in reviews are the most significant drivers of the sales decline. This provides the business with clear, data-driven, and actionable insights to guide product improvements.

High-level flow

- 01. Explore and visualize data:** Perform a descriptive analysis to understand the data and validate that the sales shortfall for a model of the product is actually true.
- 02. Extract review sentiment using generative AI-powered features:** Use the `AI.SCORE` function to extract structured sentiment scores for each product feature mentioned in customer reviews.
- 03. Create a unified feature table:** Combine the AI-generated sentiment data with product order information and user profiles (for example, age, gender, or urbanicity).
- 04. Train and evaluate a model:** Train an XGBoost model in BQML to understand how each feature contributes to sales. Evaluate the model's performance using SQL-based evaluation functions.
- 05. Explain the outcome:** Use the model's feature importance scores to explain the poor sales of the latest product version, linking them directly to sentiment in customer reviews.

Automating contract analysis

The business challenge

For legal, procurement, and finance departments in large organizations, the sheer volume of legal contracts represents a significant operational bottleneck and a hidden source of risk. Manually reviewing thousands of vendor agreements, sales contracts, and partnership deals to identify risks, track obligations, and ensure compliance is an expensive, slow, and notoriously error-prone process. A single missed renewal date can lead to unwanted service continuation, and an overlooked non-standard liability clause can expose the company to millions of dollars in unexpected risk.

The problem is one of scale and accessibility. Critical information is locked away in unstructured PDF and Word documents, making it impossible to perform portfolio-wide analyses. Legal teams cannot easily answer fundamental questions like "How many of our contracts have a termination for convenience clause?" or "Which agreements are governed by a specific jurisdiction?" without a time-consuming manual review. This reactive approach prevents the business from proactively managing its contractual risks and obligations.

The BigQuery AI approach

Transform a qualitative, manual task into a quantitative, automated one by leveraging Gemini models directly within BigQuery. By treating unstructured text documents in Cloud Storage as if they were structured data, use the power and familiarity of SQL to perform large-scale analysis.

The workflow begins by creating a remote model in BigQuery that points to a Gemini endpoint in Gemini Enterprise Agent Platform. Alternatively, the step of creating a remote model can be skipped by using the `AI.GENERATE` function and specifying the Gemini model version there. Then, use the `AI.GENERATE_TABLE` function, providing it with a target schema and a natural language prompt. This prompt instructs the model on exactly what information to extract from each contract, such as "Identify the governing law," "Extract

the contract's effective date and termination date," or "Summarize the limitation of liability clause."

The model processes each document and populates a new, structured BigQuery table based on the defined schema. This effectively turns a repository of thousands of legal documents into a queryable database. Data scientists and analysts can then use standard SQL to instantly find patterns, identify anomalies, and aggregate risk exposure across the entire contract portfolio. This enables them to build dashboards that monitor key clauses, set up alerts for upcoming renewal dates, and provide leadership with a comprehensive, data-driven view of the company's contractual landscape.

High-level flow

- 01. Access contract data in BigQuery:** Create an external BigQuery table that directly reads raw contract text files from a Google Cloud Storage bucket.
- 02. Enable generative AI in BigQuery:** Establish a connection between BigQuery and Gemini Enterprise Agent Platform and create a remote model that points to a Gemini model.
- 03. Transform text into structured data:** Use `AI.GENERATE_TABLE()` to execute a single SQL query that instructs the Gemini model to parse each document and return structured information.
- 04. Enrich data with AI summaries:** Use `AI.GENERATE()` to create a concise executive summary for each contract.
- 05. Analyze and visualize insights:** With the contract data now fully structured and enriched, use standard SQL queries to perform analysis. The results are then visualized using Plotly to create an interactive dashboard that answers key business questions, such as:
 - *What is the distribution of different contract types?*
 - *Which contracts have the highest risk scores?*
 - *What is the average risk score per contract category?*
 - *How many contracts fall under specific legal jurisdictions?*

Identifying customer segments

The business challenge

Marketing teams often need to optimize budget allocation to maximize return on investment (ROI).

A one-size-fits-all marketing strategy is inefficient, leading to wasted ad spend and low engagement, as the messaging is not relevant to all recipients.

The business problem is to develop a systematic, data-driven method for partitioning a customer base into distinct groups based on their behavior, enabling more effective and personalized marketing campaigns.

While basic segmentation using simple demographic data is straightforward, it often fails to capture the more nuanced differences in customer purchasing patterns. The analytical challenge is to move beyond these simple heuristics and identify meaningful segments based on complex behavioral data, such as purchase frequency, monetary value, and product category preferences. The goal is to produce segments that are not only statistically distinct but also interpretable and actionable for the marketing team.

The BigQuery AI approach

Combine unsupervised machine learning with generative AI to create and characterize customer segments. First, apply a k-means clustering algorithm directly within BigQuery ML to efficiently partition the entire customer dataset based on purchasing behavior.

Clustering effectively groups customers and generates a cluster ID. The second step is to automate the interpretation of these segments to provide business context. Use a generative AI function to analyze the behavioral data of customers within each cluster and programmatically generate qualitative descriptions, including concise segment name, segment summary, and tailored marketing suggestions for each segment.

High-level flow

- 01. Prepare customer data:** Consolidate and clean customer purchasing data for analysis.
- 02. Train a clustering model:** Create and train a k-means clustering model using a single SQL query in BigQuery ML to group customers into a specified number of segments.
- 03. Assign customers to segments:** Use the ML.PREDICT function to assign each customer to their respective segment.
- 04. Generate segment personas with AI:** Use the AI.GENERATE_TABLE() function to analyze the members of each cluster and automatically generate a concise segment name, segment summary, and tailored marketing suggestions for each segment.

Building an image-based home search engine

The business challenge

For e-commerce platforms, particularly those selling products where images factor into the purchase decision, like appliances, furniture, or real estate, the limitations of traditional keyword-based search create significant friction in the customer journey. A user often has a clear visual idea of the product they want to purchase but may struggle to describe it with a precise set of keywords (for example, "mid-century modern home with a big front lawn"). This mismatch between visual intent and text-based search can lead to user frustration, poor search results, and, ultimately, abandoned shopping carts.

The business objective is to create a more intuitive and effective product discovery experience that aligns with how users think visually. The technical challenge is to build a system that can accept an image as a search query and return a ranked list of the most visually similar products from a catalog of potentially millions of items. Solving this problem of visual search is critical for improving search relevance, increasing customer engagement, and driving higher conversion rates.

The BigQuery AI approach

Implement a large-scale similarity search system using vector embeddings. The approach here is to treat the problem of visual similarity as one of proximity in a high-dimensional vector space, building the entire workflow on BigQuery's native multimodal capabilities. Start by using a remote foundation model, called from a BigQuery ML SQL query, to convert each product image into a high-dimensional vector embedding. To enable fast querying over millions of vectors, create a VECTOR INDEX on the embeddings.

The search functionality is then exposed through the VECTOR_SEARCH function. When a user provides a query image, this is converted into an embedding and the VECTOR_SEARCH function is used to find the products with the closest embeddings in the indexed catalog, returning a ranked list of visually similar items in real-time.

High-level flow

- 01. Prepare multimodal product data:** Consolidate structured product information (such as descriptions or features) and unstructured data (such as images) into a BigQuery table.
- 02. Generate image embeddings:** Use a remote model in BigQuery ML to generate high-dimensional vector embeddings for each product image.
- 03. Create a vector index:** Build a vector index on the embeddings column in your BigQuery product table using the CREATE VECTOR INDEX command to enable fast approximate nearest neighbor (ANN) lookups.
- 04. Enable semantic search:** Use the VECTOR_SEARCH() function in a query. When a user submits an image, it is converted into an embedding, and this function finds the most semantically similar products based on the proximity of their embeddings.
- 05. Enhance product discovery:** Integrate the vector search results into the e-commerce platform to power 'find similar products' features and improve the overall customer search experience.

Enabling conversational analytics

The business challenge

Extracting actionable business insights from enterprise data is heavily bottlenecked by a reliance on deep SQL expertise and overworked data analyst teams. Non-technical business leaders are frequently forced to wait days or weeks for custom reports, which delays critical decision making. While basic text-to-SQL tools attempt to bridge this gap, they are often fragile in practice. They lack critical business context – such as domain-specific terminology, organization-specific glossaries, and complex filtering rules – which leads to inaccurate query results, hallucinated fields, and a general lack of trust in AI-generated answers.

The BigQuery AI approach

The Agent Catalog and conversational analytics address these hurdles directly by enabling teams to deploy highly curated, enterprise-ready conversational data agents. Rather than relying on simple, unguided translations, data teams enrich these agents with Gemini-suggested table metadata, custom system instructions, and business glossaries. Most importantly, administrators can embed verified queries (historically known as golden queries) directly into the agent. This teaches the agent your organization's exact business logic, ensures absolute accuracy in SQL generation, and allows users to query complex datasets with simple, natural language.

High-level flow

01. Create and register a custom Conversational Analytics (CA) agent: Navigate to BigQuery Studio to create and register a custom CA agent and connect it to up to 100 knowledge sources (including tables, views, graphs, or UDFs).

Note: There is no limit to the number of supported knowledge sources via the API.

02. Configure, verify, and test the CA agent: Configure Gemini-generated column descriptions, custom instructions (for example, deduplication and default limits), verified queries, and glossaries. Run profile scans to get access to sample values and other useful insights. Test the CA agent in a real-time preview panel.

03. Business users converse with the CA agent in natural language for rich insights: Once published, business users can start a conversation in Gemini Enterprise, BigQuery Studio, Looker, or programmatically via the Conversational Analytics API. The agent processes each user prompt through a structured response stream: showing its step-by-step reasoning, generating and running optimized SQL, displaying results in a table, rendering a smart chart visualization, extracting key narrative insights, and recommending logical next questions.

Preparing data with AI agents

The business challenge

Raw enterprise data is notoriously messy, fragmented, and inconsistent – often containing concatenated fields (like product names and categories merged with symbols), mixed string-and-currency columns, and mismatched date formats. Historically, preparing, transforming, and validating this data for analytics required deep SQL expertise, custom Python scripting, and specialized data engineering resources. This heavy reliance on technical teams created an operational bottleneck, leaving business practitioners waiting for weeks on backlogged IT queues just to get clean data ready for basic business intelligence.

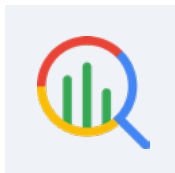
The BigQuery AI approach

Resolves this bottleneck by embedding Gemini-powered AI agents, such as Data Engineering Agent and Data Science Agent, directly within the BigQuery Studio console to democratize and automate ingestion, transformation, cleansing, and data science. Rather than requiring manual coding, the AI agents autonomously analyze raw datasets to recommend smart transformations, build data pipelines, and deploy machine learning models for predictive analytics. Data Engineering Agent lets you build, modify, and manage data pipelines to load and process data in BigQuery. With the [Data Engineering Agent](#), you can use natural language prompts to generate data pipelines from various data sources or adapt existing data pipelines to suit your data engineering needs.

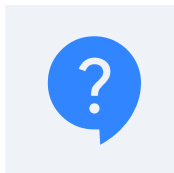
High-level flow

- 01. Launch Data Engineering Agent:** Launch the BigQuery pipelines interface, and prompt Data Engineering Agent to create a plan to analyze raw files in Google Cloud Storage, or a table schema and recommend immediate cleanses.
- 02. Apply recommended transformations:** Apply recommended transformations to isolate key metrics, extract attributes using pattern recognition, and enrich the dataset by executing AI-assisted joins with secondary tables.
- 03. Automate the data pipeline:** Data Engineering Agent standardizes the data pipeline by fixing inconsistent date formats and sets up data quality assertions (such as identifying missing values). Finally, an optimization skill proposes column pruning or pushdown operations, the clean data is written to a final destination table, and the entire pipeline can be scheduled to run automatically.

Get started on your data science journey



[Try BigQuery for free*](#)



[Contact us for expert advice](#)

*Store 10 GB of data and run up to 1 TiB of queries for free per month. New customers also get \$300 in free credits to try BigQuery and other Google Cloud products.



Contributors

Manoj Gunti
Senior Product Marketing Manager,
Google Cloud

Joe Malone
Product Manager, Google Cloud

Diogo Kato
Cloud Data Consultant,
Professional Services, Google Cloud

Brad Miro
Developer Advocate, Google Cloud

Manan Goel
Group Product Manager, Google Cloud

Vasiya Krishnan
Product Manager, Google Cloud

Jeff Nelson
Developer Advocate, Google Cloud

Geeta Banda
Head of Outbound Product
Management, Google Cloud

Shan Kulandaivel
Group Product Manager, Google Cloud

Vaibhav Sethi
Group Product Manager, Google Cloud

Tim Bezold
Product Manager, Google Cloud

Jing Jing Long
Director Engineering, Google Cloud

Alicia Williams
Developer Advocate, Google Cloud

