



Generative AI with Vertex AI: Build a customer chatbot

Coursera Project Network

Generative AI with Vertex AI: Build a customer chatbot	1
Learning Objectives	2
Project Overview	2
Welcome to Your Guided Project!	2
Meet Your Instructor	3
Guided Project Structure	3
Earn Your Certificate	3
Task 1: Summarize customer calls in Vertex AI console	3
Project Outcome, Learning Objectives	5
Project Scenario	5
Your Role : Data Scientist	5
Task 1 Summzier customer calls in Vertex AI console	6
Go to cloud google verex ai	6
Go to google console	7
Enable all Vertex APIS	8
Generative AI Studio → Click Try out	8
Three Category : langage , Speech, Vision	9
Select Language in Vertex AI	10
Go to Summary —> Support chat summary prompt template	11
Prompt example	12
View code button	12
Region support	13
Support Select LLM Model	14
Temperature parameter control degree of randomness in token selection	15
Advance	16
Token limit setting	16
Add Stop sequence : avoid toxic content	17
Go to the link https://community.spiceworks.com/t/call-transcript-from-a-banking-call-center/929216	18
Copy the chat conversion	18
Place and submit in vertex AI	19
result	20
Given Prompt instruction	21
Task 2: Summarize customer calls in Google Colab	22
Go to Colab	23
Create new Notebook with name “Building vertex AI chatbot for financial institution	24
Install google-cloud-aiplatform library	25
Restart runtime	26
Go to vertex AI → view code	27
Python code	28
Python colab code for Vetrex AI	29
Initial vertex model	29
Copy python code	30
Paste and run to colab	31

Prompt allow this notebook access your google credentials (not working – not grain authorization)	32
Link for describe Vertex AI authorization	32
If you're working in Colaboratory, run the following command in a Colab cell to authenticate:	32
Task Summary	33
Practice 1: Summarize article in Google Colab	33
Practice 1 Activity	34
Demo	35
Initial vertex model	36
Add digital	36
Task 3: Analyze customer calls with Vertex AI console	37
Use previous code	38
Open Classify help tickets prompt	39
Input /output example	39
Text section for input	40
Train data , example input → output label	41
Download bank transaction —> copy row 15 transaction	42
Paster into test input ticket	43
Result classify intop credit card serive	43
Copy row 39	44
Task Summary	45
Task 4: Classify customer calls with Google Colab	46
Click View code from previous exercise	47
Copies code	48
Predict	48
Update banking_dataset.csv file to colab	49
Load CSV file	50
First 5 row	50
Add create predict category in dataframe by LLM	51
Task Summary	52
Practice 2: Classify IT customer request tickets	53
Practice 2 Activity	53
Load CSV file	54
Classification label	55
Result	55
Task 5: Build a simple customer service chatbot with Vertex AI	56
Context “you’re a customer service representative of a financial institution dedicated to keeping customer happy”	
59	
Type : “Hello”	60
In Python Colab	61
Task summary	63
Cumulative Task	63
Import Vertexai ChatModel , load chatmodel	65
Set chatbot parameters , start chat setup (context, example (input output text pair)	66
TextGenerate model Summarize the conversion ,	67
Use TextGenerate model classificate call Categories	68
Assess Your Knowledge	69

Learners will learn how to leverage NLP particularly text classification, summarization, and a chatbot through both the Google Cloud Vertex AI console interface and with the Google Cloud APIs in a Jupyter Notebook to gain insights into their customer service communications.

Learning Objectives

-
- Summarize customer support calls

- Classify customer support calls
- Build a simple interactive customer service chatbot that provides information to customers on request

Project Overview

Welcome to Your Guided Project!

In this 1 hour long project-based course, you will learn how to leverage Vertex AI to summarize, classify and build a chatbot for a financial institution.

We will analyze customer support calls and build a customer support chatbot. We will learn how to use the Vertex AI console in addition to building our own programs in Google Colabs calling the various APIs and pre-trained models. This course is aimed at learners who are looking to get into leveraging cloud language AI solutions through Vertex AI.

The prerequisites are having a Google cloud account, have some basic knowledge of Google Cloud Vertex AI console (the UI for this is simple and very intuitive even if a learner has not encountered it before), a basic working knowledge of python including how to call functions, how to import python modules, how to set variables, how to install python modules, how API calls work in python and how to open the Google Colab website.

Be sure to download the following dataset which are also linked in the resources section of the project.

1. Banking dataset

[banking_dataset](#)

[CSV File](#)

2. IT customer tickets

[customertickets](#)

[CSV File](#)

Keep in mind that if don't have Google Cloud setup, you can follow this [link](#) to help you get set up.

Take note that if this is the first time you're using Vertex AI, when you open the console, you would be required to enable the Vertex AI cloud API before you can use.

Meet Your Instructor

Hi my name is Emmanuel Acheampong and I will be taking you through this Guided Project. I am a co-founder and Head of AI at roboMUA(robo-Makeup Artist) an AI company and an AI conference speaker at conferences like Ai4, AI Dev World, etc.

Guided Project Structure

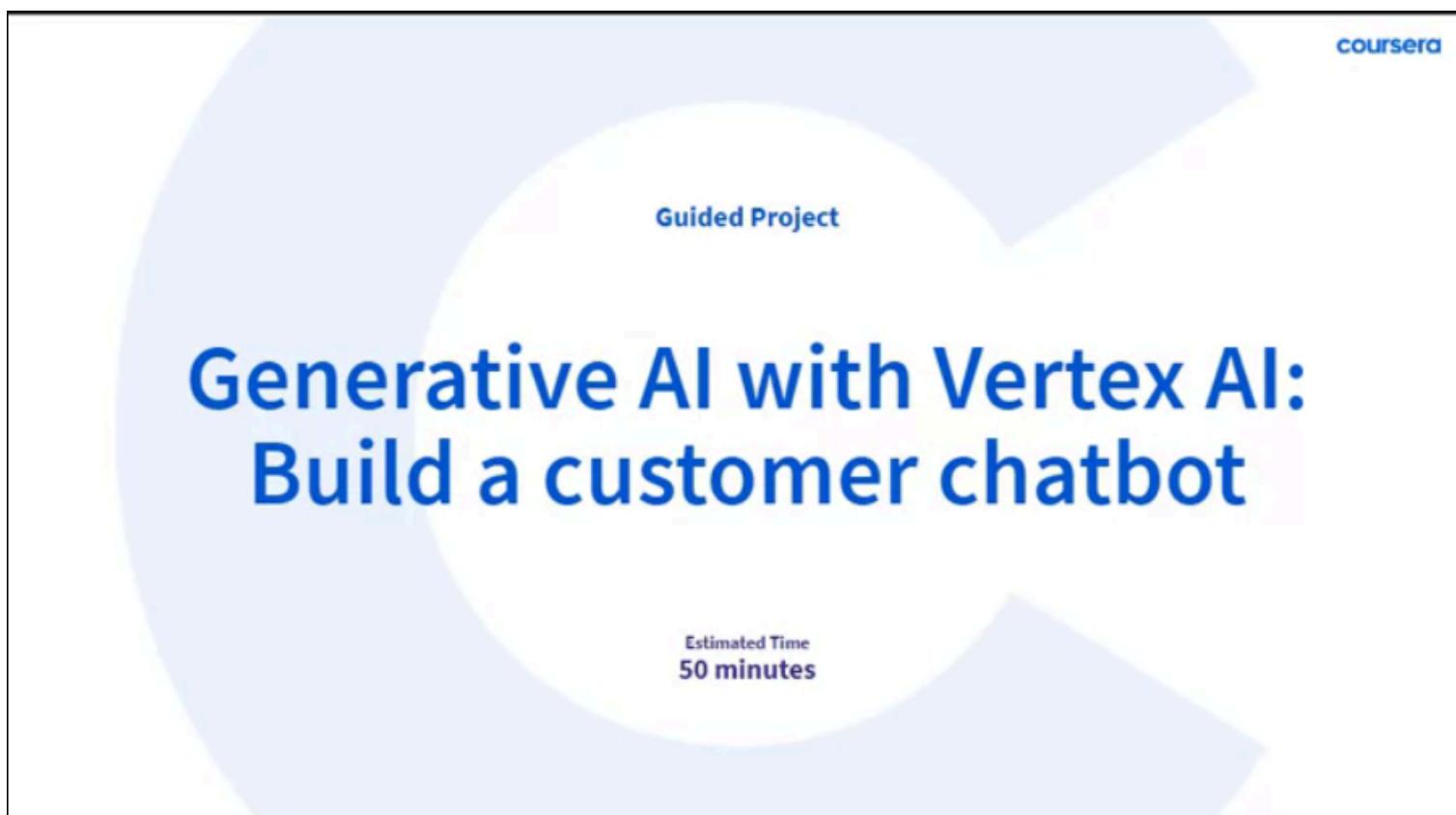
This Guided Project is divided into **4** parts:

1. **Project Overview:** This introductory reading material.
2. **Hands-on Project:** The hands on project that we will work on together
3. **Graded Quiz:** The final assignment that you need to pass in order to finish the project successfully.
4. **Learner Survey:** Tell us what you thought about this guided project!

Earn Your Certificate

After you have completed the hands-on project, you will be able to assess your knowledge using a graded assignment on Coursera. Score higher than 80% to [earn your certificate!](#)

Task 1: Summarize customer calls in Vertex AI console



Project Outcome, Learning Objectives

INTRODUCTION coursera



Project Outcome

Leverage NLP—particularly sentiment analysis, text classification, summarization, extraction and a chatbot through both the Google Cloud Vertex AI console interface and with the Google Cloud APIs in a Jupyter Notebook.

Learning Objectives

- Summarize customer support calls.
- Classify customer support calls.
- Build a simple interactive customer service chatbot that provides information to customers on request

particularly sentiment analysis,

Project Scenario

Your Role : Data Scientist



coursera

Project Scenario

Your Role: Data Scientist

A leading financial institution has realized that a growing number of customers have been leaving their wealth management division to join a rival firm. The executives are trying to figure out how to stop this from happening but the only data they have is the emails and calls with customer service personnel. Learners will take on the role of a data scientist who can leverage cloud NLP to help executives understand on a high level the kind of information and pain points customers are sharing with customer service reps through calls and emails before they decide to leave the firm. A Jupyter notebook that can analyze this data and even serve as a chatbot to communicate with a customer to ensure their needs are being met will help executives get to the bottom of why customers are leaving and how best to prevent that.

given the influx of

Task 1 Summarize customer calls in Vertex AI console

Task 1

Summarize customer calls in Vertex AI console

We will introduce the Vertex AI console, do a small tour of the console platform before navigating to the generative AI language dashboard for summarization.

Our goal is simply to familiarize

Go to cloud google verex ai

Vertex AI

Innovate faster with enterprise-ready generative AI

Vertex AI offers everything you need to build and use generative AI—from AI solutions, to Search and Conversation, to 100+ foundation models, to a unified AI platform.

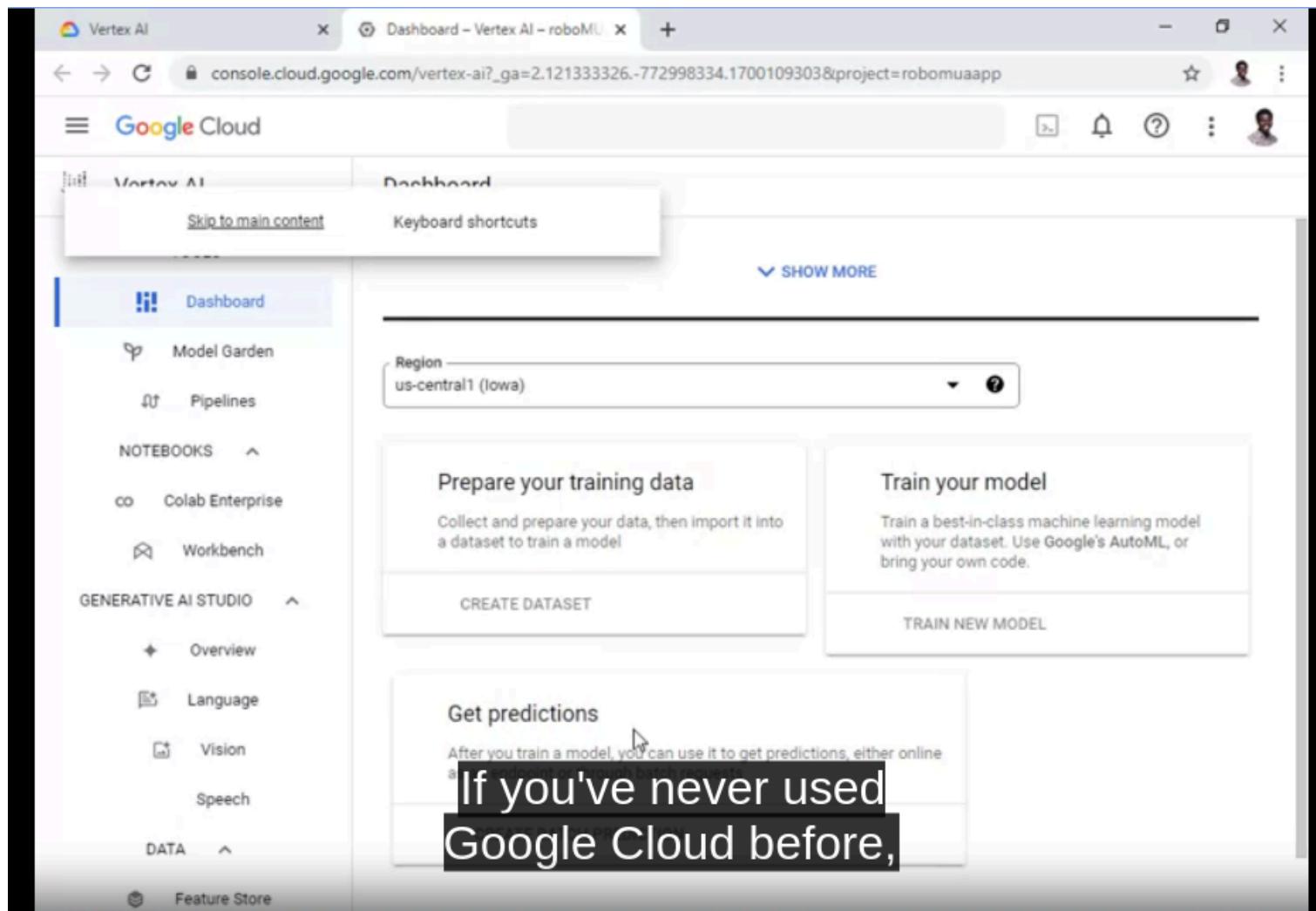
Go to console

Product highlights

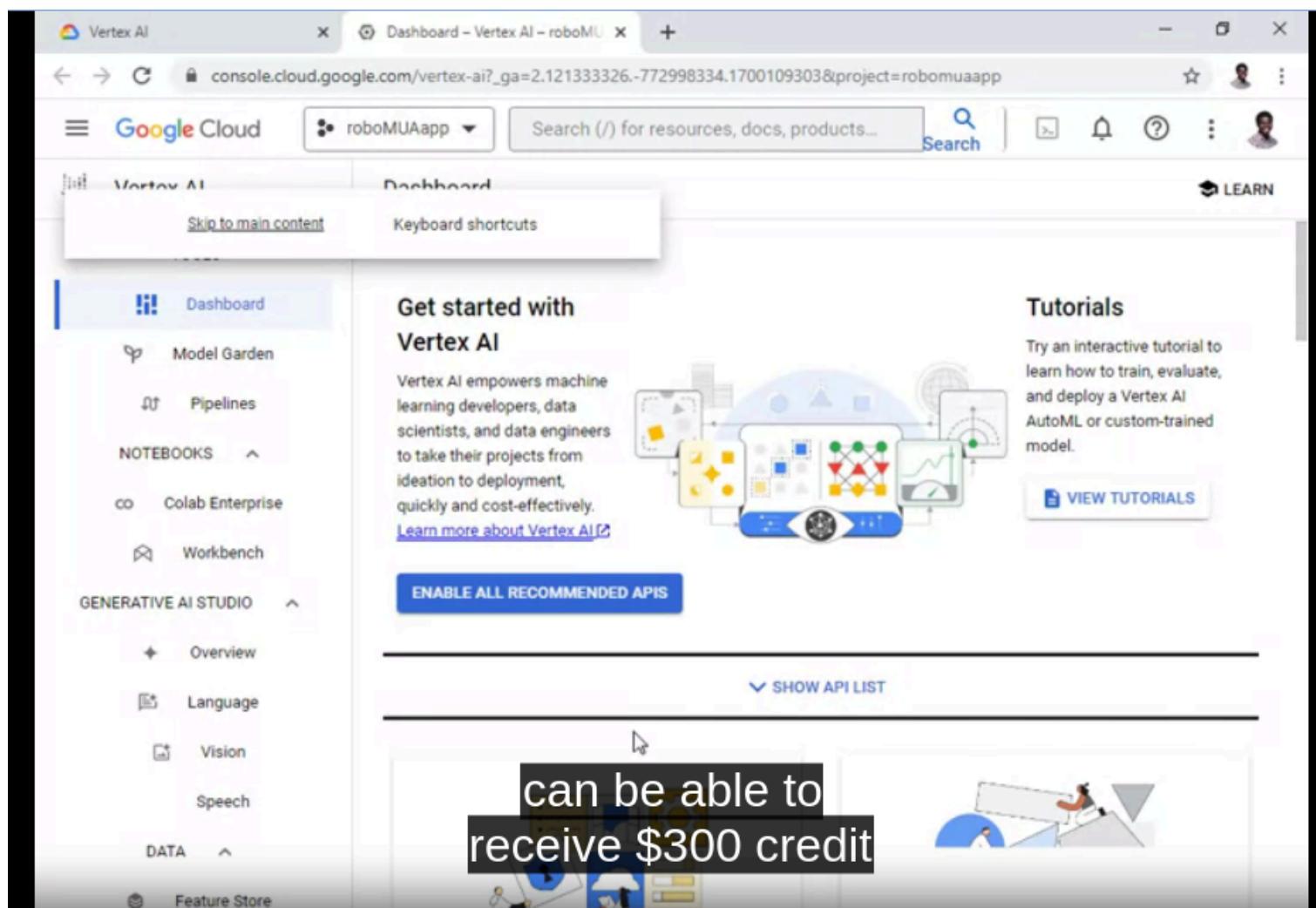
- Train, test, and tune ML models on a single platform
- Build generative AI apps quickly

Waiting for cache...

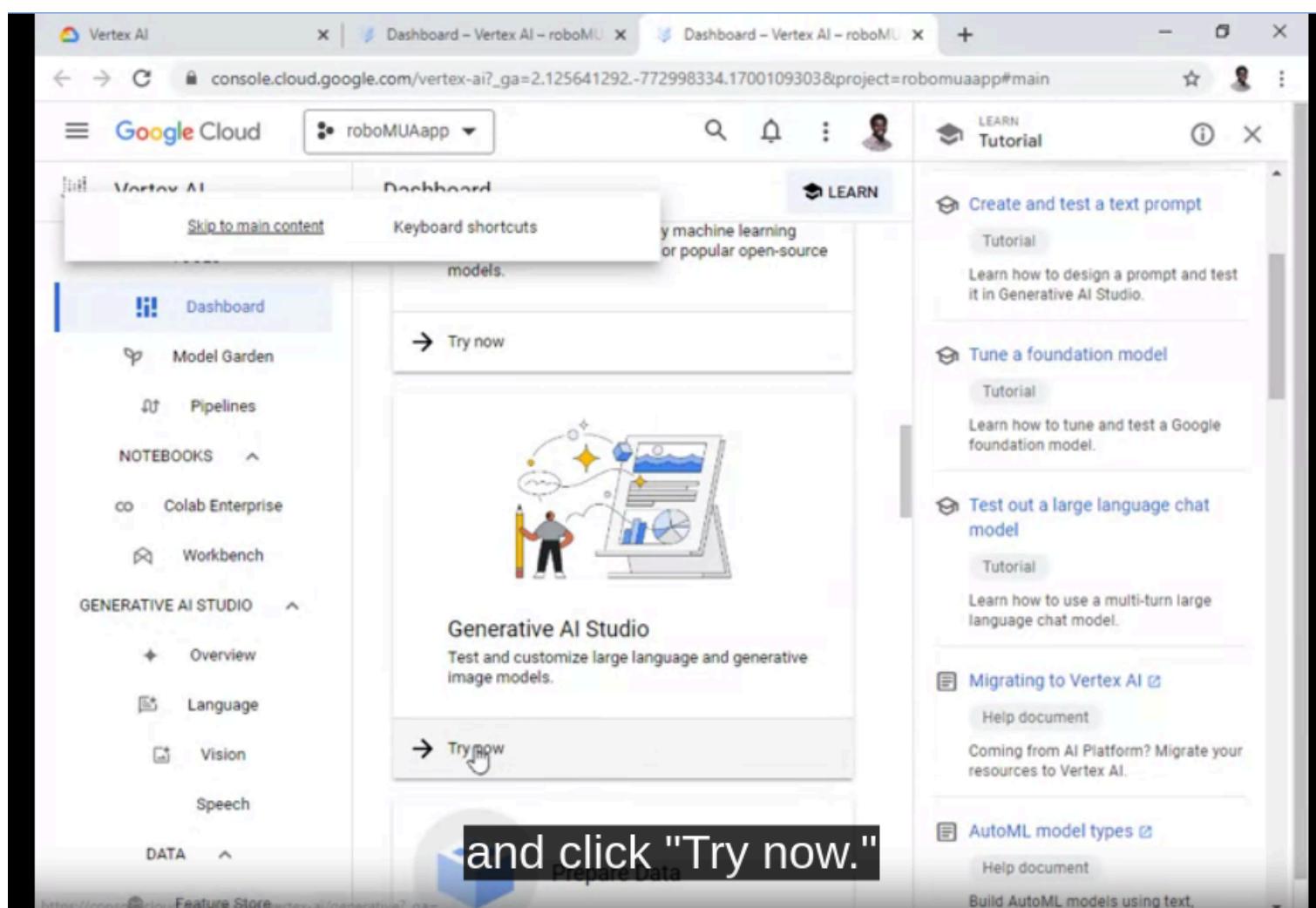
Go to google console



Enable all Vertex APIs



Generative AI Studio → Click Try out



Three Category : language , Speech, Vision

The screenshot shows the Google Cloud Vertex AI Generative AI Studio interface. On the left, there is a sidebar with navigation links: Dashboard, Model Garden, Pipelines, Notebooks (Colab Enterprise, Workbench), Generative AI Studio (Overview, Language, Vision, Speech), and Data (Feature Store). The main content area features three main sections: Language, Speech, and Vision. The Language section is highlighted with a large black overlay containing the text "We have language, speech, and vision." Below the Language section, there is a "Write text" input field and a "Generate image" button. The Speech section describes converting speech into text or synthesizing speech from text using Google's Universal Speech Model (USM). The Vision section is partially visible at the bottom.

Vertex AI

Google Cloud

roboMUAapp

Search (/) for resources, docs, produ... Search

Skip to main content Keyboard shortcuts

Dashboard

Model Garden

Pipelines

NOTEBOOKS

Colab Enterprise

Workbench

GENERATIVE AI STUDIO

Overview

Language

Vision

Speech

DATA

Feature Store

Language

Speech

Vision

We have language, speech, and vision.

Write text

Generate image

OPEN

OPEN

Select Language in Vertex AI

The screenshot shows the Google Cloud Vertex AI Language interface. The left sidebar includes sections for Dashboard, Model Garden, Pipelines, Notebooks (Colab Enterprise), Workbench, Generative AI Studio (Overview, Language, Vision, Speech), Data (Feature Store), and Language. The main area is titled "Create a new prompt" and contains two main sections: "Generate text" and "Start a conversation". The "Generate text" section has "TEXT PROMPT" and "CODE PROMPT" buttons. The "Start a conversation" section has "TEXT CHAT" and "CODE CHAT" buttons. A central banner says "Take some time, pause".

Vertex AI

Dashboard – Vertex AI – roboMU... Language – Vertex AI – roboMU...

Google Cloud robomuaapp Search (/) for resources, docs, produ... Search

Vertex AI Language

Skip to main content Keyboard shortcuts TUNING

Dashboard Model Garden Pipelines NOTEBOOKS Colab Enterprise Workbench GENERATIVE AI STUDIO Overview Language Vision Speech DATA Feature Store

Create a new prompt

Generate text Start a conversation

Design prompts for tasks relevant to your business use case including code generation. [View tutorial](#)

Have a freeform chat with the model, which tracks what was previously said and responds based on context. [View tutorial](#)

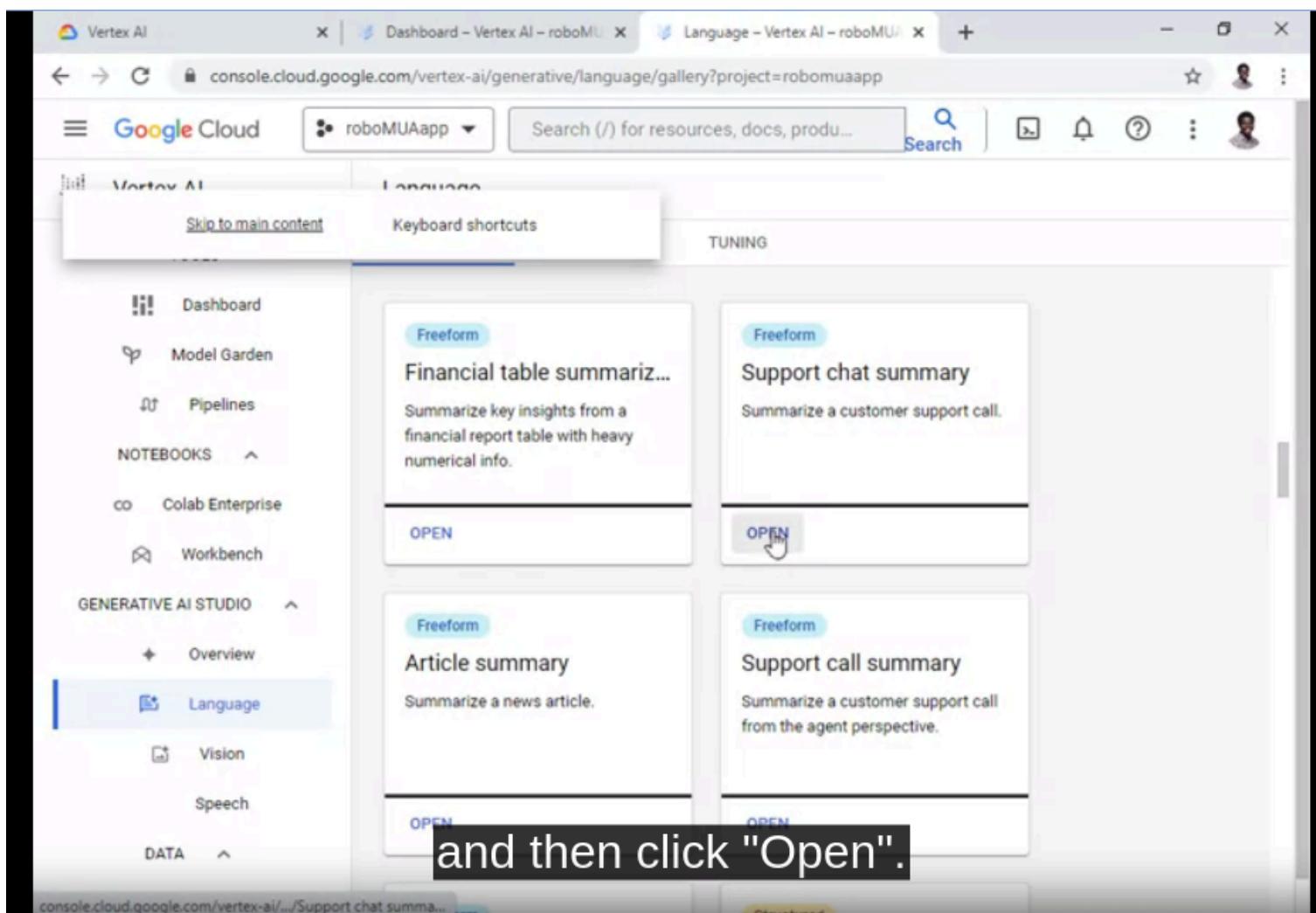
TEXT PROMPT CODE PROMPT TEXT CHAT CODE CHAT

Explore more

Access the API Take some time, pause Tune a model

Tune a model so it's your use case, then endpoint to get precise prompt design. [View tutorial](#)

Go to Summary —> Support chat summary prompt template



Prompt example

The screenshot shows the Google Cloud Vertex AI Language interface. The left sidebar includes sections for Dashboard, Model Garden, Pipelines, Notebooks, Colab Enterprise, Workbench, Generative AI Studio (with Overview, Language, Vision, and Speech options), Data (with Feature Store), and Feature Store. The main area is titled "Support chat summary" and contains a "Prompt" section with the following text:

Summarize the following conversation between a service rep and a customer in a few sentences. Use only the information from the conversation.

Service Rep: How may I assist you today?
Customer: I need to change the shipping address for an order.
Service Rep: Ok, I can help you with that if the order has not been fulfilled from our warehouse yet. But if it has already shipped, then you will need to contact the shipping provider. Do you have the order ID?

To the right of the prompt are configuration settings:

- We want your [feedback](#).
- Region: us-central1 (Iowa)
- Model: text-bison (latest)
- Temperature: 0.2

At the bottom, there are "SUBMIT" and "RESET PARAMETERS" buttons, and a note: "The model will generate a response after you click Submit". A large watermark text "Here you'll see a prompt and a response section." is overlaid across the middle of the page.

View code button

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with options like Dashboard, Model Garden, Pipelines, Notebooks, Colab Enterprise, Workbench, Generative AI Studio (with Overview, Language, Vision, Speech), Data, and Feature Store. The 'Language' option under Generative AI Studio is currently selected. The main area is titled 'Support chat summary' and contains a 'Prompt' section with a conversation between a service rep and a customer. To the right, there are settings for Region (us-central1 (Iowa)), Model (text-bison (latest)), and Temperature (set to 0.2). Buttons for 'SUBMIT' and 'RESET PARAMETERS' are at the bottom. A large watermark-like text 'you're going to see a Save tab, View Code tab.' is overlaid across the middle of the screen.

Region support

This screenshot is similar to the one above, showing the Vertex AI interface. The 'Region' dropdown in the settings panel is highlighted with a cursor, indicating it's being interacted with. The same watermark text 'which a subscription ID is currently located.' is overlaid at the bottom.

Support Select LLM Model

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with options like Dashboard, Model Garden, Pipelines, Notebooks, Colab Enterprise, Workbench, Generative AI Studio (Overview, Language, Vision, Speech), Data, and Feature Store. The main area is titled "Support chat summary". It contains a "Prompt" section with a conversation between a service rep and a customer about changing a shipping address. Below the prompt, a large text box displays the generated response: "reasoning since they are basically just". At the bottom of this box, it says "The model will generate a response after you click Submit". To the right, there's a "Region" dropdown set to "us-central1 (Iowa)", a "Model" section with a filter, and a list of available models under "PaLM 2" and "Codey (PaLM 2)". A feedback message at the top right says "We want your [feedback](#)".

We want your [feedback](#).

Region: us-central1 (Iowa)

Model:

Filter Type to filter

PaLM 2

- text-bison (latest) [PREVIEW](#)
- Latest
- text-bison-32k (latest) [PREVIEW](#)
- Latest
- text-bison@001
- Best value

Codey (PaLM 2)

- code-bison (latest) [PREVIEW](#)
- Content processed through Vertex AI is assessed against a list of safety attributes. Confidence scores for these attributes are returned in API responses [but are not visible in the UI](#).

reasoning since they
are basically just

The model will generate a response after you click Submit

Temperature parameter controls degree of randomness in token selection

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with various AI services: Dashboard, Model Garden, Pipelines, Notebooks, Colab Enterprise, Workbench, Generative AI Studio (with Overview and Language selected), Vision, Speech, Data (with Feature Store), and Feature Store.

The main area is titled "Support chat summary" and contains a "Prompt" section. The prompt text is:

```
Summarize the following conversation between a service rep and a customer in a few sentences. Use only the information from the conversation.  
Service Rep: How may I assist you today?  
Customer: I need to change the shipping address for an order.  
Service Rep: Ok, I can help you with that if the order has not been fulfilled from our warehouse yet. But if it has already shipped, then you will need to contact the shipping provider. Do you have the order ID?  
Customer: Yes, it's 88986367.
```

To the right of the prompt, there are configuration options:

- Region: us-central1 (Iowa)
- Model: text-bison (latest)
- Temperature: A slider set to 0.2

Below these options are "SUBMIT" and "RESET PARAMETERS" buttons. A large watermark text "controls the degree of randomness in token selection." is overlaid across the middle of the interface.

Advance

Token limit setting

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with sections like 'TOOLS' (Dashboard, Model Garden, Pipelines), 'NOTEBOOKS' (Colab Enterprise, Workbench), and 'GENERATIVE AI STUDIO' (Overview, Language, Vision, Marketplace). The 'Language' section is currently selected. The main area is titled 'Support chat summary' and contains a 'Prompt' section with a text input field containing a conversation transcript. To the right of the prompt are several configuration options: 'Region' set to 'us-central1 (Iowa)', 'Model' set to 'text-bison (latest)', 'Temperature' set to 0.2, and a 'Token limit' slider set to 256. Below these are 'Advanced' settings for 'Top-K' (set to 40) and a 'Sampling temperature' slider set to 0.8. A large watermark 'text output from one prompt.' is overlaid across the middle of the interface.

Add Stop sequence : avoid toxic content

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with sections for TOOLS (Dashboard, Model Garden, Pipelines), NOTEBOOKS (Colab Enterprise, Workbench), and GENERATIVE AI STUDIO (Overview, Language, Vision, Marketplace). The 'Language' option under Generative AI Studio is currently selected. The main workspace is titled 'Support chat summary'. In the 'Prompt' section, there's a conversation transcript:

```
Summarize the following conversation between a service rep and a customer in a few sentences. Use only the information from the conversation.  
Service Rep: How may I assist you today?  
Customer: I need to change the shipping address for an order.  
Service Rep: Ok, I can help you with that if the order has not been fulfilled from our warehouse yet. But if it has already shipped, then you will need to contact the shipping provider. Do you have the order ID?  
Customer: Yes, it's 88986367.
```

To the right of the prompt, there are several configuration options:

- Top-K: Set to 40
- Top-P: Set to 0.8
- Max responses: Set to 1
- Streaming responses: A checkbox labeled "Print responses as they're generated"
- Safety filter threshold: Set to "Block few"

At the bottom of the interface, there's a large black bar with the text "Finally, at the bottom," overlaid. Below this bar, a note says "The model will generate a response after you click Submit". There are also "SUBMIT" and "RESET PARAMETERS" buttons, along with a link to "REPORT INAPPROPRIATE RESPONSES".

Go to the link

<https://community.spiceworks.com/t/call-transcript-from-a-banking-call-center/929216>

Copy the chat conversion

The screenshot shows a web browser window with three tabs: "Vertex AI", "Vertex AI - roboMUAapp ~ Go...", and "Call Transcript from a Banking C...". The main content area displays a chat transcript:

I can fix that for you, could I have your customer identification number.
32948322
Your customer pin number..
Um, don't remember, I hardly ever use it.
That's fine, I am just going to have ask some security questions to verify you are the account holder. Date of birth?
1/1/1975
Favourite chess opening..
Sicilian
Most disturbing childhood memory..
Almost drowning in a pond after going in because I wouldn't go in because I co...

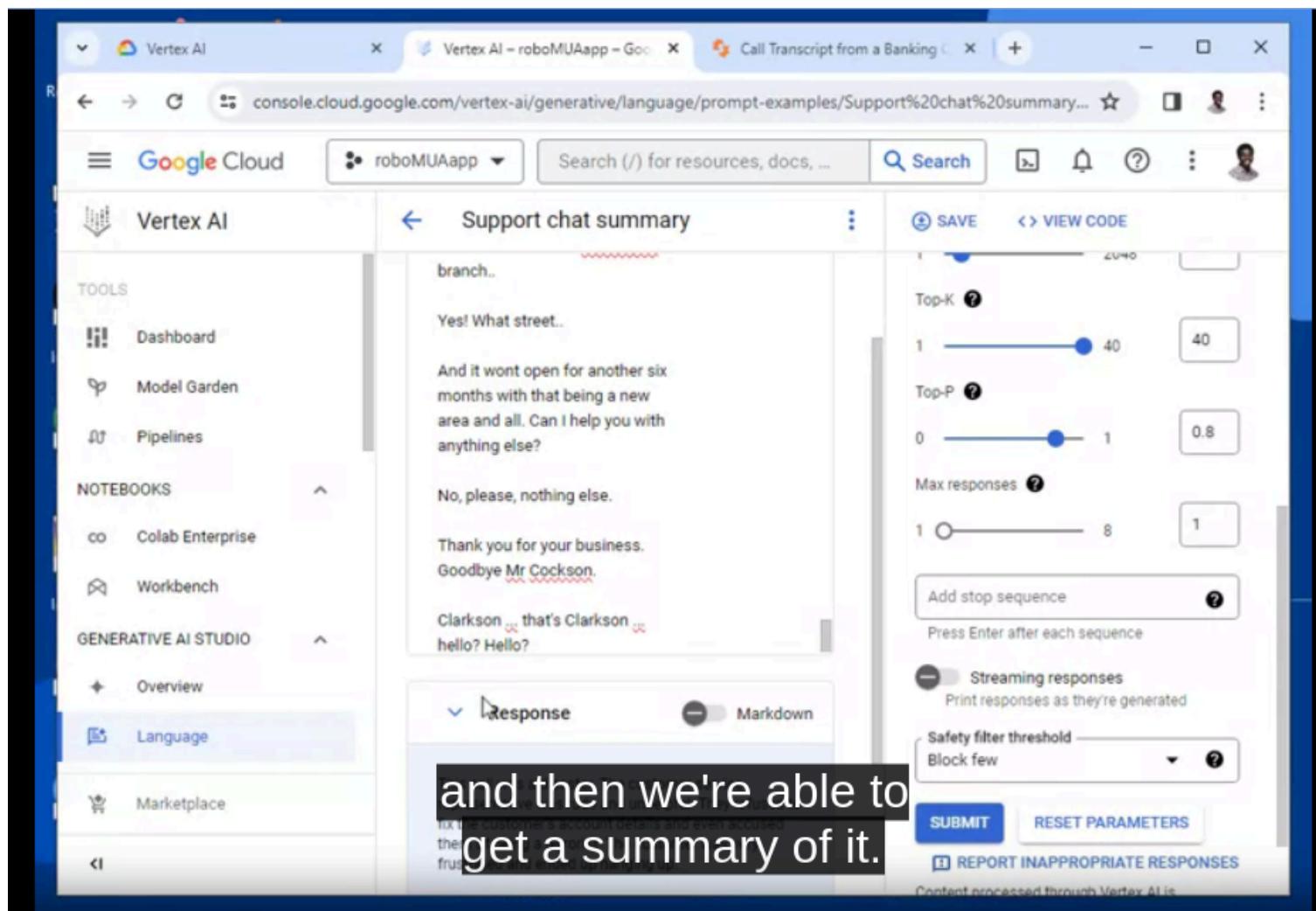
A large black rectangular box covers the bottom portion of the transcript, containing the text: "Scroll down starting from high all the way to the end,".

On the right side of the browser window, there is a sidebar titled "Popular Topics in Data Management" with several listed items:

- WARNING [CTL10301, CTL10300] Creating analysis-only dataset
- Have you explored edge computing in your IT projects?
- Data transfer tools
- loop through based on different condition in informatica
- Which technology has had the biggest energy-saving impact on your l...

[View all topics](#)

Place and submit in vertex AI



result

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with sections for TOOLS (Dashboard, Model Garden, Pipelines), NOTEBOOKS (Colab Enterprise, Workbench), and GENERATIVE AI STUDIO (Overview, Language, Marketplace). The main area is titled "Support chat summary" and contains a transcript of a conversation:

```
months with that being a new area and all. Can I help you with anything else?  
No, please, nothing else.  
Thank you for your business.  
Goodbye Mr Cockson.  
Clarkson 🇬🇧 that's Clarkson 🇬🇧 hello? Hello?
```

Below the transcript, there's a "Response" section containing a generated summary:

This call was a disaster. The customer service representative was rude and unhelpful. They refused to fix the customer's account details and even accused them of being a terrorist. The customer was very frustrated and ended up hanging up.

Overlaid on the response section is a large, bold text: "and then we're able to get a summary of it."

On the right side of the interface, there are several configuration sliders and buttons:

- Top-K: Set to 40
- Top-P: Set to 0.8
- Max responses: Set to 1
- Add stop sequence: A text input field with placeholder "Press Enter after each sequence".
- Streaming responses: A toggle switch that is off.
- Safety filter threshold: A dropdown menu set to "Block few".
- Buttons at the bottom: SUBMIT (blue), RESET PARAMETERS, and REPORT INAPPROPRIATE RESPONSES.

A small note at the bottom of the response section says: "Model may display inaccurate or offensive information that".

Given Prompt instruction

The screenshot shows the Google Cloud Vertex AI interface. In the top left, it says "Vertex AI" and "Vertex AI - roboMUAapp - Go...". The URL is "console.cloud.google.com/vertex-ai/generative/language/prompt-examples/Support%20chat%20summary?proj...". The main area is titled "Support chat summary". There are two tabs: "FREEFORM" (selected) and "STRUCTURED". A "CLEAR PROMPT" button is at the top right of the prompt area. The prompt text is:

I summarize the following conversation between a service rep and a customer in a few sentences. Use only the information from the conversation.

Service Rep: How may I assist you today?
Customer: I need to change the shipping address for an order.
Service Rep: Ok, I can help you with that if the order has not been fulfilled from our warehouse yet. But if it has already shipped, then you will need to contact the shipping provider. Do you have the order ID?
Customer: Yes, it's 88986367.
Service Rep: One minute please while I pull up your order information.
Customer: No problem.

Model may display inaccurate or offensive info
are sup...

On the right side, there are settings for "Region" (us-central1 (Iowa)), "Model" (text-bison (latest)), "Temperature" (0.2), and "Token limit" (2048). Below these are "SUBMIT" and "RESET PARAMETERS" buttons, and a link to "REPORT INAPPROPRIATE RESPONSES". A note at the bottom states: "Content processed through Vertex AI is assessed against a list of safety attributes. Confidence scores for these attributes are returned in API responses but are not visible in this UI." A large watermark in the center of the page reads "an instruction for any summary you're".

The screenshot shows a "Task Summary" page from Coursera. At the top, it says "Task Summary" and "coursera".

Task Objective

Summarize customer calls in Vertex AI console

Key Takeaways

- Sign up for Google Cloud If you don't have it set up. See the Resources section or Getting Started section for help.

Congratulations.
Now let's summarize

Task 2: Summarize customer calls in Google Colab

The screenshot shows a white rectangular card against a black background, representing a task card on a learning platform. In the top left corner, the word "TASK OBJECTIVE" is written in small blue capital letters. In the top right corner, the "coursera" logo is visible. A cursor arrow is positioned in the top right corner of the card. The main title "Task 2" is centered at the top in large blue capital letters. Below it, the subtitle "Summarize customer calls in Google Colab" is displayed in bold blue capital letters. Underneath the subtitle, a descriptive text reads: "Leveraging the Vertex APIs in Google Colab to summarize call transcripts". At the bottom of the card, there is a text box containing the message: "Welcome back. In the previous task,". This text is partially obscured by a large yellow curved arrow that starts from the bottom right and sweeps across the card towards the center.

TASK OBJECTIVE

coursera

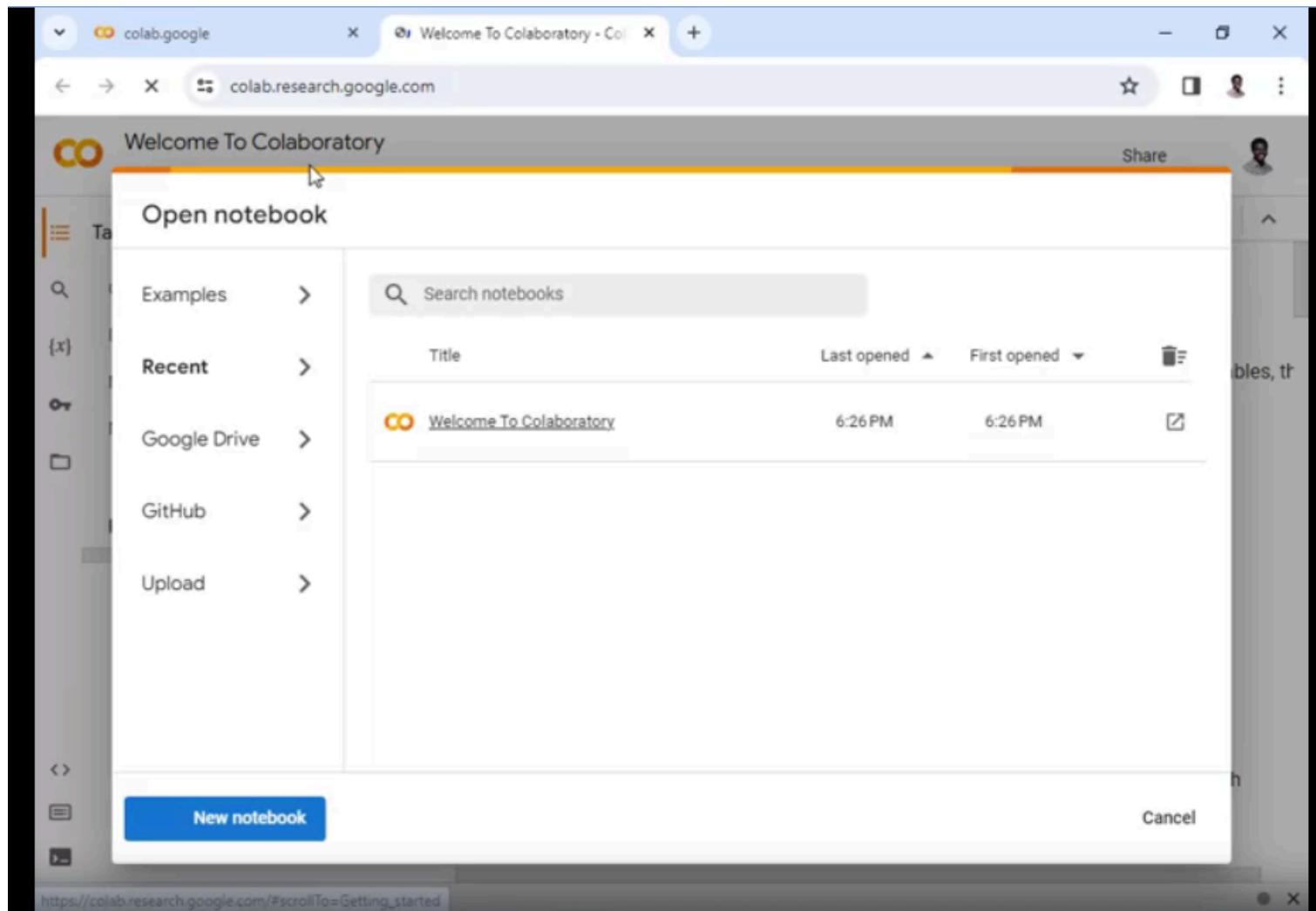
Task 2

Summarize customer calls in Google Colab

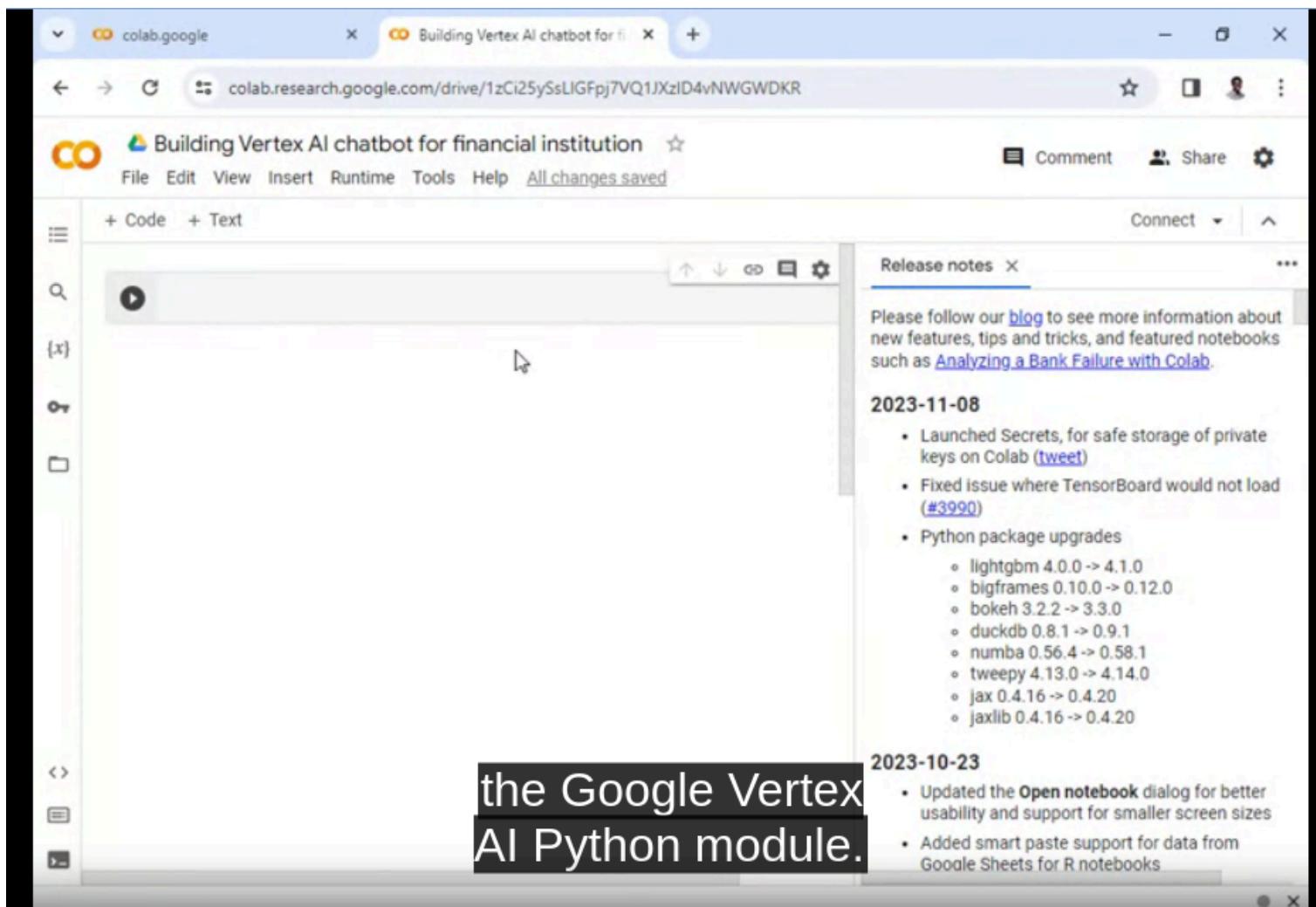
Leveraging the Vertex APIs in Google Colab to summarize call transcripts

Welcome back. In the previous task,

Go to Colab



Create new Notebook with name “Building vertex AI chatbot for financial institution



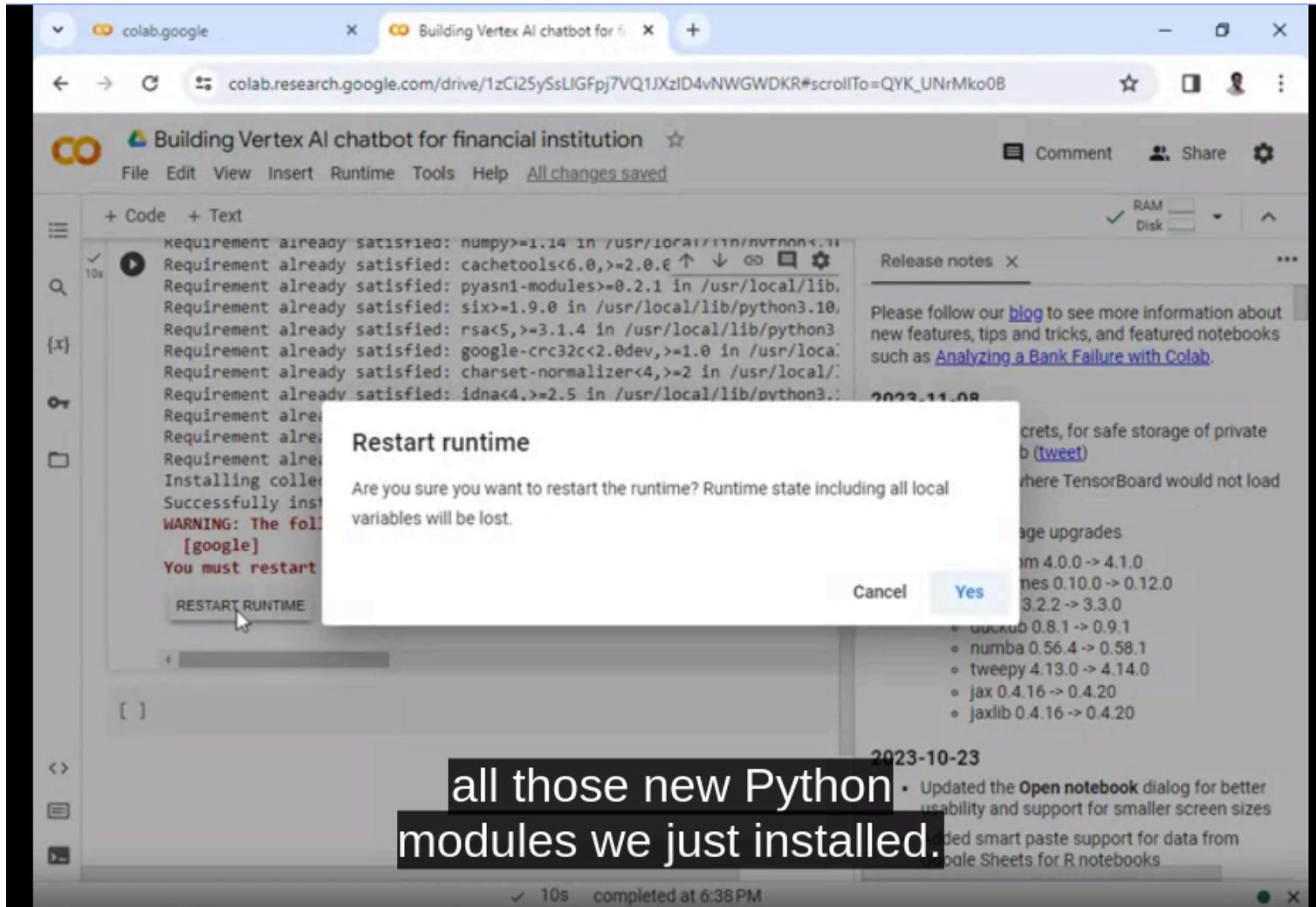
Install google-cloud-aiplatform library

The screenshot shows a Google Colab notebook interface. The title bar says "Building Vertex AI chatbot for financial institution". The main area displays a code cell with the following content:

```
[1] Requirement already satisfied: typing-extensions>=4.2.0 in /usr/local/
Collecting zope-interface>=5 (from Twisted>=13.1.0->vertex)
  Downloading zope.interface-6.1-cp310-cp310-manylinux_2_5_x86_64.many.
  247.1/247.1 kB 23.1 MB/s
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-pi
Requirement already satisfied: cffi>=1.12 in /usr/local/lib/python3.10/
Requirement already satisfied: idna>=2.5 in /usr/local/lib/python3.10/
Requirement already satisfied: setuptools in /usr/local/lib/python3.10/
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/
Building wheels for collected packages: vertex
  Building wheel for vertex (setup.py) ... done
  Created wheel for vertex: filename=Vertex-0.3.1-py3-none-any.whl size
  Stored in directory: /root/.cache/pip/wheels/4a/52/81/ac01030d8b73b8e
Successfully built vertex
Installing collected packages: incremental, zope-interface, hyperlink,
Successfully installed Twisted-23.10.0 automat-22.10.0 constantly-23.10.0
...
!pip install google-cloud-aiplatform --upgrade
```

The status bar at the bottom indicates "Executing (1s)". A tooltip is overlaid on the screen with the text "just '!pip install google...'" and "Shift Enter to run again.".

Restart runtime



Go to vertex AI → view code

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with categories like Vertex AI, Colab Enterprise, Workbench, GENERATIVE AI STUDIO (with Overview, Language, Vision, Speech), DATA (with Feature Store, Datasets, Labeling tasks, Marketplace), and a bottom section for AI-generated images. The main area is titled "Support call summary" and shows a "Prompt" section with a message about a branch location. To the right, there are configuration options for Region (us-central1 (Iowa)), Model (text-bison (latest)), Temperature (set to 0.2), and a "SUBMIT" button. A large watermark in the center says "we can see view code, click View Code." Below the configuration area, there's a note about reporting inappropriate responses.

Vertex AI – roboMUAapp – Goo colab.google Building Vertex AI chatbot for fo... - X

console.cloud.google.com/vertex-ai/generative/language/prompt-examples/Support%20call%20summary?proj...

Google Cloud robоМUAApp Search (/) for resources, docs, prod... Search

Vertex AI Support call summary

Prompt

... shows your ridiculous address is right around the corner from our Chrichfield branch..

Yes! What street..

And it wont open for another six months with that being a new area and all. Can I help you with anything else?

No, please, nothing else.

Thank you for your business. Goodbye Mr Cockson.

Clarkson ... that's Clarkson ... hello?
Hello?

This call was a disaster. The customer service representative was rude and unhelpful. They refused to fix the customer's...

we can see view code, click View Code.

REPORT INAPPROPRIATE RESPONSES

Content processed through Vertex AI is assessed against a list of safety attributes. Confidence scores for these attributes are returned in API responses but are not visible in the UI.

SAVE <> VIEW MODE

We want your feedback.

Region: us-central1 (Iowa)

Model: text-bison (latest)

Temperature: 0.2

SUBMIT RESET PARAMETERS

Python code

The screenshot shows a browser window with three tabs: "Vertex AI - roboMUAapp - Goo", "colab.google", and "Building Vertex AI chatbot for f...". The main content area is titled "View code" and has tabs for "PYTHON", "PYTHON COLAB", and "CURL". The "PYTHON" tab is selected. On the left, there's a sidebar with "Vertex AI" and "GENERATIVE AI STUDIO" sections, and a "Language" item under GENERATIVE AI STUDIO is highlighted. The main content area contains a script demonstrating how to request a model response using the Vertex AI SDK for Python. The script imports vertexai and its language_models module, initializes the project, sets parameters for generating text, and uses a TextGenerationModel to predict responses to user input. A large black overlay box covers the bottom right portion of the code area, containing the text "the Vertex AI SDK is used in Python.".

```
import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="robomuaapp", location="us-central1")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 1024,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
model = TextGenerationModel.from_pretrained("text-bison")
response = model.predict(
    """Hi, you've called Swish Bank, how may I help you?

I'm a customer of your bank and I have some errors on my account details.

What type of errors?

Well my name is spelt incorrectly for a start, it's spelt c-l-a-r-k-e-s-o-n, it shouldn't have an e. /

I can fix that for you, could I have your customer identification number.

32948322
Your cust"""
)
```

the Vertex AI SDK
is used in Python.

Python colab code for Vtex AI

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with categories like Vertex AI, Colab Enterprise, Workbench, GENERATIVE AI STUDIO (Overview, Language, Vision, Speech), DATA (Feature Store, Datasets, Labeling tasks, Marketplace), and a bottom section for COLAB. The main area has tabs for View code, PYTHON, PYTHON COLAB, and CURL. The PYTHON tab is selected. It contains two sections: 1. Instructions to install the Vertex AI SDK for Python and restart the Colab notebook runtime. 2. A code snippet demonstrating how to use the vertexai library to initialize a model and generate text. A large black box with white text overlays the code area, stating: "From the code we can see that we first authenticate".

```
1. Install the Vertex AI SDK for Python. After installing you'll be prompted to restart your Colab notebook runtime.  
!pip install "shapely<2.0.0"  
!pip install google-cloud-aiplatform --upgrade  
  
2. Use the following code in your notebook to request a model response  
import vertexai  
from vertexai.language_models import TextGenerationModel  
  
vertexai.init(project="robomuaapp", location="us-central1")  
parameters = {  
    "candidate_count": 1,  
    "max_output_tokens": 1024,  
    "temperature": 0.2,  
    "top_p": 0.8,  
    "top_k": 40  
}  
model = TextGenerationModel.from_pretrained("text-bison")  
response = model.predict(  
    """Hi, you've called Swish Bank, how may I help you?  
  
I'm a customer of your bank and I have some errors on my account details.  
  
What type of errors?"""
```

From the code we can see that we first authenticate

Initial vertex model

The screenshot shows the Google Cloud Vertex AI interface. The sidebar and tabs are identical to the previous screenshot. The main area displays the same Python code for generating text. A large black box with white text overlays the code area, stating: "We then initialize the Vertex AI, set the parameters,".

```
1. Install the Vertex AI SDK for Python. After installing you'll be prompted to restart your Colab notebook runtime.  
!pip install "shapely<2.0.0"  
!pip install google-cloud-aiplatform --upgrade  
  
2. Use the following code in your notebook to request a model response  
import vertexai  
from vertexai.language_models import TextGenerationModel  
  
vertexai.init(project="robomuaapp", location="us-central1")  
parameters = {  
    "candidate_count": 1,  
    "max_output_tokens": 1024,  
    "temperature": 0.2,  
    "top_p": 0.8,  
    "top_k": 40  
}  
model = TextGenerationModel.from_pretrained("text-bison")  
response = model.predict(  
    """Hi, you've called Swish Bank, how may I help you?  
  
I'm a customer of your bank and I have some errors on my account details.  
  
What type of errors?"""
```

We then initialize the Vertex AI, set the parameters,

Copy python code

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with categories like Vertex AI, Colab Enterprise, Workbench, GENERATIVE AI STUDIO (Overview, Language, Vision, Speech), DATA (Feature Store, Datasets, Labeling tasks), and Marketplace. The 'Language' section is currently selected. The main area is titled 'View code' and contains instructions for requesting a model response with a Colaboratory notebook. It includes two sections: 1. Instructions for installing the Vertex AI SDK for Python and upgrading dependencies. 2. A code snippet for generating text using a TextGenerationModel. The code imports vertexai and from vertexai.language_models import TextGenerationModel. It initializes the model with project="robomuaapp" and location="us-central1". Parameters are set to {"candidate_count": 1, "max_output_tokens": 1024, "temperature": 0.2, "top_p": 0.8, "top_k": 40}. The model is then used to predict a response. A large black box highlights the text 'the copy button at the top right of the code section,' with a cursor pointing towards the 'Copy' button.

Use this script to request a model response with a Colaboratory notebook.

1. Install the [Vertex AI SDK](#) for Python. After installing you'll be prompted to restart your Colab notebook runtime.

```
!pip install "shapely<2.0.0"  
!pip install google-cloud-aiplatform --upgrade
```
2. Use the following code in your notebook to request a model response

```
import vertexai  
from vertexai.language_models import TextGenerationModel  
  
vertexai.init(project="robomuaapp", location="us-central1")  
parameters = {  
    "candidate_count": 1,  
    "max_output_tokens": 1024,  
    "temperature": 0.2,  
    "top_p": 0.8,  
    "top_k": 40  
}  
model = TextGenerationModel.from_pretrained("text-bison")  
response = model.predict(  
    """Hi, you've called Swish Bank, how may I help you?  
  
I'm a customer of your bank and I have some errors on my account details.  
What type of errors?""")
```

the copy button at the top right of the code section,

Paste and run to colab

The screenshot shows a Google Colab notebook interface. The title bar reads "Building Vertex AI chatbot for financial institution". The left sidebar has icons for file operations like New, Open, Save, and Delete. The main area shows a conversation transcript and some Python code.

Transcript:

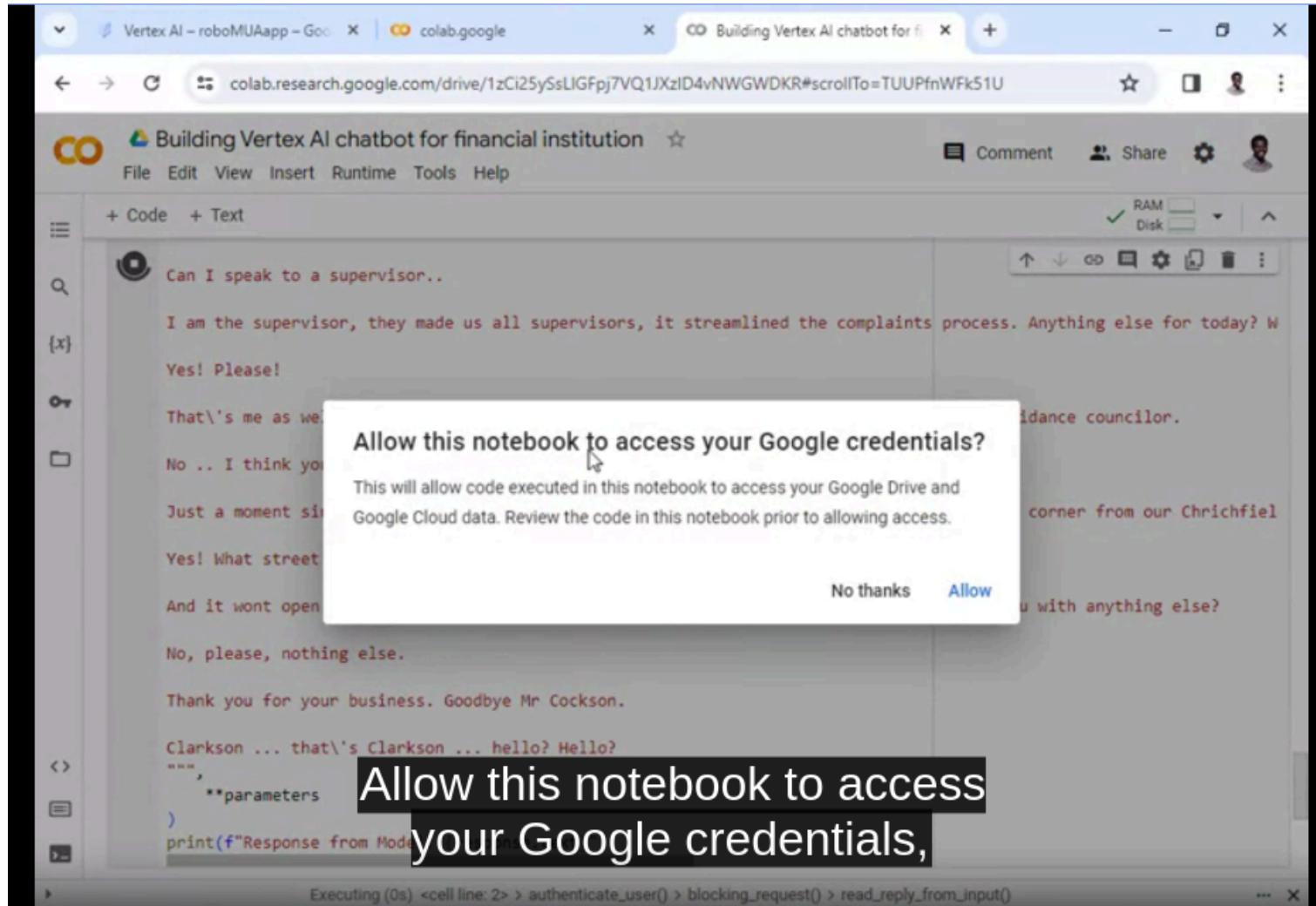
- Can I speak to a supervisor..
- I am the supervisor, they made us all supervisors, it streamlined the complaints process. Anything else for today? W
- Yes! Please!
- That's me as well! How can I help you? I'm also my performance manager and my career guidance councilor.
- No .. I think you've done enough. I need to get to a bank to try and get some cash..
- Just a moment sir, yes my branch finder shows your fictitious address is right around the corner from our Chrichfiel
- Yes! What street..
- And it wont open for another six months with that being a new area and all. Can I help you with anything else?
- No, please, nothing else.
- Thank you for your business. Goodbye Mr Cockson.
- Clarkson ... that's Clarkson ... hello? Hello?

Code:

```
"""
    **parameters
)
print(f"Response from Model: {response.text}")
```

At the bottom, a status bar indicates "✓ 6s completed at 8:03PM".

Prompt allow this notebook access your google credentials (not working – not grain authorization)



Link for describe Vertex AI authorization

https://cloud.google.com/vertex-ai/generative-ai/docs/multimodal/sdk-for-gemini/gemini-sdk-overview-reference?_gl=1*sbz3k8*_ga*MTc5NjA1NjAxOC4xNzEzNTAyNjM3*_ga_WH2QY8WWF5*MTcxMzk3NDc4Ny44LjEuMTcxMzk3NTYwMy4wLjAuMA..&_ga=2.233109598.-1796056018.1713502637&_gac=1.187240026.1713974787.CjwKCAjw26KxBhBDEiwAu6KXt41Zw45Ysi5Z9vxCqNNlbOsIpOChsjs-09IvfMnFy8qir1X20gWPURoCSrAQAvD_BwE

If you're working in Colaboratory, run the following command in a Colab cell to authenticate:

```
from google.colab import auth  
auth.authenticate_user()
```

Task Summary

The screenshot shows a 'Task Summary' section from a Coursera course. At the top right is the Coursera logo. Below it, the title 'Task Summary' is displayed in blue. To the left of the title is a yellow icon of a mountain with a flag at the peak. The main content area has a light gray background. It contains two sections: 'Task Objective' and 'Key Takeaways'. The 'Task Objective' section includes a brief description: 'Learners will create a Google colab notebook cell to use google cloud APIs for support call summary'. The 'Key Takeaways' section lists two bullet points: 'Shapely is not a required python module to install for Google Colab' and 'The parameters are configurable and can be changed to get better results.' A small cursor icon is visible on the right side of the page.

Task Summary

Task Objective

Learners will create a Google colab notebook cell to use google cloud APIs for support call summary

Key Takeaways

- Shapely is not a required python module to install for Google Colab
- The parameters are configurable and can be changed to get better results.

our goal was to summarize

Practice 1: Summarize article in Google Colab

coursera

GUIDED PROJECT

Practice 1 Activity



This task is optional and ungraded.
The goal is to check your understanding.

Welcome back, in this optional task we
are going to be summarizing a news article

Practice 1 Activity

coursera

Practice 1 Activity

- Import the necessary python modules
- Initialize vertex ai
- Set the pretrained model
- Insert data input
- Call the vertex AI modules to summarize the text.

 **Things to Note**

- You need to sign in with a Google account for a free version of colab.
- Be sure to install all the necessary python modules in Google Colab.
- Restart runtime session if the installed modules are not working

 **Pro Tip**

- ★ If you're stuck check out the code generated in the vertex AI console.

(Note to the student: Please note that this practice test is not graded. It is intended to help you understand the concepts covered in the course.)

For this particular practice test you can either open a new Google Colab notebook so

Demo

The screenshot shows a Google Colab notebook titled "Building Vertex AI chatbot for financial institution". The code cell contains the following Python code:

```
from google.colab import auth as google_auth
google_auth.authenticate_user()

import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="robomuaapp", location="us-central1")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 1024,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
model = TextGenerationModel.from_pretrained("text-bison")
response = model.predict(
    """
    JAMIE DIMON: First, I got a big bus behind me. And when we do th
    LESLIE TAYLOR: That's right. And when we do that, we do it in a
    """
)
```

A large text overlay covers the bottom half of the code cell, reading:

So after importing the necessary Python modules, initializing Vertex AI,

Initial vertex model

```
google_auth.authenticate_user()

import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="robomuaapp", location="us-central1")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 1024,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
model = TextGenerationModel.from_pretrained("text-bison")
response = model.predict(
    """
    LESLIE PICKER: Hey, Kelly, thank you so much. And thank you, Jamie, for being here. So you've got a big bus behi
    JAMIE DIMON: First, we love these bus trips. It's like 12 years, we try to go off the beaten path. And when we do th
    PICKER: What have you learned so far this year?
    DIMON: Okay, so this is for
    PICKER: Well, let's
    """
)
```

setting the parameters,
importing the necessary LLM.

11s completed at 9:32PM

Add digital

```
PICKER: What do you think the price would ultimately be?
DIMON: Well I said, there will be higher interest rates. And I don't know what it's gonna do to the economy.
PICKER: In terms of the consumer, do you think they fully absorb that stimulus yet and absorbed the period of low in
DIMON: Yeah. That's what I'm talking about. The storm clouds down the road - you know, fiscal spending, QT, higher r
PICKER: Do you think the Fed's done a good job?
DIMON: They did a good job early on. I think it is quite obvious it was too much. It was just their job. You know, f
PICKER: Let's talk about the Fed in terms of regulation. Last week, we saw the Basel three end game, Basel 4. Some p
DIMON: Is there some way someone can actually quite that group down over there. Sorry.
PICKER: Is that a metaphor -
DIMON: I don't know.
PICKER: You know, higher capital requirements for you all, what do you think is the overall impact from these new ru
And then predicting on
DIMON: First of all, I just want to say that I'm very disappointed with these rules. I mean, I've alwa
PICKER: Speaking of the recent banking crisis, you know, you letter to shareholders in April "the current
```

11s completed at 9:32PM

Task 3: Analyze customer calls with Vertex AI console

The image shows a screenshot of a presentation slide from Coursera. The slide has a black header bar with the word "TASK OBJECTIVE" on the left and the Coursera logo on the right. Below the header, the title "Task 3" is displayed in blue. Underneath the title, the subtitle "Classify customer calls Vertex AI console" is shown in a dark blue font. A small gray text below the subtitle reads "Using vertex AI LLM for setting context and categories of calls." At the bottom of the slide, there is a large black rectangular area containing white text that says "In this task, we're going to be classifying". A yellow curved arrow starts from the top right of the slide and points towards the text in the black box.

TASK OBJECTIVE

coursera

Task 3

Classify customer calls Vertex AI console

Using vertex AI LLM for setting context and categories of calls.

In this task, we're going to be classifying

Use previous code

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with 'Vertex AI' selected under 'TOOLS'. Below it are sections for 'NOTEBOOKS' (Colab Enterprise, Workbench), 'GENERATIVE AI STUDIO' (Overview, Language, Vision, Marketplace), and a 'CLOSE' button. The main area is titled 'View code' and contains instructions and code snippets for interacting with a model. A large black box highlights the text 'the view code panel where we'.

Use this script to request a model response with a Colaboratory notebook.

1. Install the [Vertex AI SDK](#) for Python. After installing you'll be prompted to restart your Colab notebook runtime.

```
!pip install "shapely<2.0.0"  
!pip install google-cloud-aiplatform --upgrade
```
2. Use the following code in your notebook to request a model response

```
from google.colab import auth as google_auth  
google_auth.authenticate_user()  
  
import vertexai  
from vertexai.language_models import TextGenerationModel  
  
vertexai.init(project="robomuaapp", location="us-central1")  
parameters = {  
    "candidate_count": 1,  
    "max_output_tokens": 1024,  
    "temperature": 0.2,  
    "top_p": 0.8,  
    "top_k": 40  
}  
model = TextGenerationModel.from_pretrained("text-bison")  
response = model.predict(  
    """Summarize the following conversation from the Agent's perspective:  
Agent: Thank you for calling Google Cloud Support. How may I assist you today?  
  
Customer: Hi, I am trying to create a Google Cloud account and use the free credits, but I am not sure if  
Agent: Sure, I can definitely help you with that. May I know if you already have a Google account?""")
```

the view code panel where we

Open Classify help tickets prompt

The UI for classifying calls has

Input /output example

and the context and examples reflect that,

Text section for input

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with sections for TOOLS (Dashboard, Model Garden, Pipelines), NOTEBOOKS (Colab Enterprise, Workbench), and GENERATIVE AI STUDIO (Overview, Language, Vision, Marketplace). The main area is titled "Classify help tickets". It features a "Test" section with a table. The first row contains the input "xxxx thanks in advance for your help" and an empty output row. Below this is a "Batch Testing" section with tabs for INPUT (Ticket) and OUTPUT (Markdown). A large text overlay in the center says "The test session has examples of inputs but no outputs." To the right, there are configuration options: Region (us-central1 (Iowa)), Model (text-bison (latest)), Temperature (set to 0.2), and buttons for SUBMIT and RESET PARAMETERS. At the bottom, there's a note about content moderation: "Content processed through Vertex AI is assessed against a list of safety attributes. Confidence scores for these attributes are returned in API responses but are not visible in the UI".

The test session has examples of inputs but no outputs.

xxxx thanks in advance for your help

Write an input to add a new row

Write an output to add a new row

Test

INPUT Ticket

OUTPUT Markdown

Batch Testing

my grand son

Model responses will

fund clear my

alleviate bank

Content processed through Vertex AI is assessed against a list of safety attributes. Confidence scores for these attributes are returned in API responses but are not visible in the UI.

Train data , example input → output label

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with 'Vertex AI' selected. Under 'GENERATIVE AI STUDIO', 'Language' is highlighted. The main area is titled 'Classify help tickets'. It has an 'INPUT' section labeled 'Ticket' containing the text: 'my grand son give me check for i deposit it into my chase account after fund clear my chase bank closed my account never paid me my money they said taking by my'. To the right is an 'OUTPUT' section with a placeholder 'Model responses will appear here'. On the far right, configuration options include 'Region: us-central1 (Iowa)', 'Model: text-bison (latest)', 'Temperature: 0.2', and buttons for 'SUBMIT' and 'RESET PARAMETERS'. A large black box with white text overlaid says: 'As you can see, the model responses will appear here.'

Download bank transaction → copy row 15 transaction

	call_transcript
1	call_transcript
2	Customer: Hello, I'm calling to check on the status of my loan application.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID.
3	Customer: Hello, I'm calling to inquire about your investment products.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
4	Customer: Hello, I'm calling to transfer money to my daughter's account.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID.
5	Customer: Hello, I'm calling to dispute a charge on my credit card statement.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID.
6	Customer: Hello, I'm calling to ask about your mortgage rates.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
7	Customer: Hello, I'm calling to open a business account.\nBank: Hi, thank you for calling [bank name]. Can I have your name and business name, please?
8	Customer: Hello, I'm calling to ask about your travel insurance products.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID.
9	Customer: Hello, I'm calling to ask about your student loan rates.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
10	Customer: Hello, I'm calling to report a lost ATM card.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
11	Customer: Hello, I'm calling to inquire about your small business loan products.\nBank: Hi, thank you for calling [bank name]. Can I have your name and business name, please?
12	Customer: Hello, I'm calling to ask about your personal loan rates.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
13	Customer: Hello, I'm calling to ask about your home equity loan products.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
14	Customer: Hello, I'm calling to ask about your car loan rates.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
15	Customer: Hello, I'm calling to ask about your credit card rewards programs.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
16	Customer: Hello, I'm calling to ask about your mobile banking app.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
17	Customer: Hello, I'm calling to ask about your international banking services.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
18	Customer: Hello, I'm calling to ask about your investment advice services.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
19	Customer: Hello, I'm calling to ask about your business banking products.\nBank: Hi, thank you for calling [bank name]. Can I have your name and business name, please?
20	Customer: Hello, I'm calling to inquire about your wealth management services.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
21	Customer: Hello, I'm calling to ask about your retirement planning services.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
22	Customer: Hello, I'm calling to ask about your insurance products.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
23	Customer: Hello, I'm calling to ask about your tax preparation services.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?
24	Customer: Hello, I'm calling to ask about your retirement planning services.\nBank: Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?

Paster into test input ticket

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with sections like 'MODEL DEVELOPMENT' (Training, Experiments, Metadata), 'DEPLOY AND USE' (Model Registry, Online prediction, Batch predictions, Vector Search), and 'MANAGE' (Ray on Vertex AI, Marketplace). The main area is titled 'Classify help tickets'. It has two input fields: 'Write an input to add a new row' and 'Write an output to add a new row'. Below these is a table with columns 'INPUT' and 'OUTPUT'. In the INPUT column, there's a placeholder 'Ticket'. In the OUTPUT column, the first row shows 'Bank account services'. On the right, there are settings for 'Region' (us-central1 (Iowa)), 'Model' (text-bison (latest)), and 'Temperature' (set to 0.2). Buttons for 'SUBMIT' and 'RESET PARAMETERS' are at the bottom. A note about reporting inappropriate responses is visible.

Result classify intop credit card serive

This screenshot is from the same interface as the previous one, showing the 'Classify help tickets' page. The input field contains a ticket about a credit card. The output field shows the result: 'Credit card'. The rest of the interface, including the sidebar, model selection, and parameters, is identical to the first screenshot.

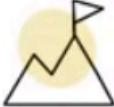
Copy row 39

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with sections for MODEL DEVELOPMENT (Training, Experiments, Metadata), DEPLOY AND USE (Model Registry, Online prediction, Batch predictions, Vector Search), and MANAGE (Ray on Vertex AI, Marketplace). The main area is titled "Classify help tickets" and contains a "Test" section. In the "Test" section, there's an "INPUT" field containing a customer message and an "OUTPUT" field showing the predicted category "Loans and Mortgages". To the right, there are configuration options for the test: Region (us-central1 (Iowa)), Model (text-bison (latest)), Temperature (set to 0.2), and buttons for SUBMIT and RESET PARAMETERS. A large watermark text "Let me see the output as loans and mortgages as well too." is overlaid across the middle of the interface.

Task Summary

The screenshot shows a 'Task Summary' section from a Coursera course. At the top right is the Coursera logo. Below it, the title 'Task Summary' is displayed in blue. To the left of the title is a yellow icon depicting a mountain with a flag at the peak. The main content area has a white background with rounded corners. It contains two sections: 'Task Objective' and 'Key Takeaways'. The 'Task Objective' section includes a yellow icon of a document with a checkmark. The 'Key Takeaways' section includes a yellow icon of a document with a checkmark. A large black bar at the bottom contains the text 'bank and phone calls in the Vertex AI console.'

Task Summary

 **Task Objective**

Learners will create a Google colab notebook cell to use the Google Cloud APIs to classify customer calls.

 **Key Takeaways**

- The classified labels in the training set are ones that will show for any input test. In other words any input data will be classified into one of the classified labels unless an unknown label is added.

bank and phone calls in the Vertex AI console.

Task 4: Classify customer calls with Google Colab

The screenshot shows a white rectangular card with rounded corners, set against a black background. In the top left corner, the word "TASK OBJECTIVE" is written in small, uppercase letters. In the top right corner, the "coursera" logo is visible. The main title "Task 4" is centered at the top in a large, bold blue font. Below it, the subtitle "Classify customer calls with Google Colab" is displayed in a smaller blue font. A descriptive text follows: "Using Google Colab and vertex AI python modules for setting context and categories of calls." At the bottom of the card, there is a large, bold, black rectangular box containing the text: "we classified customer calls using the Vertex AI console platform." Two yellow curved arrows originate from the right side of this text box and curve upwards towards the top edge of the card.

TASK OBJECTIVE

coursera

Task 4

Classify customer calls with Google Colab

Using Google Colab and vertex AI python modules for setting context and categories of calls.

we classified customer calls using
the Vertex AI console platform.

Click View code from previous exercise

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with sections like 'Vertex AI', 'TOOLS' (Dashboard, Model Garden, Pipelines), 'NOTEBOOKS' (Colab Enterprise, Workbench), and 'GENERATIVE AI STUDIO' (Overview, Language, Vision, Marketplace). The 'Language' section is currently selected. A central modal window titled 'View code' is open, containing instructions and code snippets. The 'PYTHON COLAB' tab is highlighted.

Use this script to request a model response with a Colaboratory notebook.

1. Install the [Vertex AI SDK](#) for Python. After installing you'll be prompted to restart your Colab notebook runtime.

```
!pip install "shapely<2.0.0"  
!pip install google-cloud-aiplatform --upgrade
```
2. Use the following code in your notebook to request a model response

```
from google.colab import auth  
auth.authenticate_user()  
  
import vertexai  
from vertexai.language_models import TextGenerationModel  
  
vertexai.init(project="robomuaapp", location="us-central1")  
parameters = {  
    "candidate_count": 1,  
    "max_output_tokens": 256,  
    "temperature": 0.2,  
    "top_p": 0.8,  
    "top_k": 40  
}  
model = TextGenerationModel.from_pretrained("text-bison")  
response = model.predict(  
    "***Multi-choice problem: Define the category of the ticket?  
Categories:  
- Credit card  
- Bank account services  
- Loans and Mortgages
```

hit VIEW CODE, go to the PYTHON COLAB.

Copies code

The screenshot shows the Google Cloud Vertex AI interface. On the left, there's a sidebar with 'Vertex AI' selected under 'TOOLS'. Below it are sections for 'Dashboard', 'Model Garden', 'Pipelines', 'NOTEBOOKS' (with 'Colab Enterprise' and 'Workbench' listed), 'GENERATIVE AI STUDIO' (with 'Overview' and 'Language' listed), and 'Marketplace'. The main area is titled 'View code' and contains two snippets. The first snippet is for installing the Vertex AI SDK for Python:

```
!pip install "shapely<2.0.0"
!pip install google-cloud-aiplatform --upgrade
```

The second snippet shows how to use the Vertex AI Language Model API in Python:

```
import vertexai
from vertexai.language_models import TextGenerationModel

vertexai.init(project="robomuaapp", location="us-central1")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 256,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
model = TextGenerationModel.from_pretrained("text-bison")
response = model.predict(
    """Multi-choice problem: Define the category of the ticket?
Categories:
- Credit card
- Bank account services
- Loans and Mortgages"""
)
print(response.text)
```

A large black box highlights the text: "Since we've already copied the important modules and".

Predict

The screenshot shows a Google Colab notebook titled 'Building Vertex AI chatbot for financial institution'. The left sidebar shows a 'Files' section with a 'sample_data' folder. The main area is a code editor with several ticket examples and their predicted categories:

```
"""Multi-choice problem: Define the category of the ticket?
Categories:
- Credit card
- Bank account services
- Loans and Mortgages

Ticket: I lost my credit card numbered 12345. Can you help with deactivating it?
Category: Credit card

Ticket: I would like to change the address associated with my account. I have !
Category: Bank account services

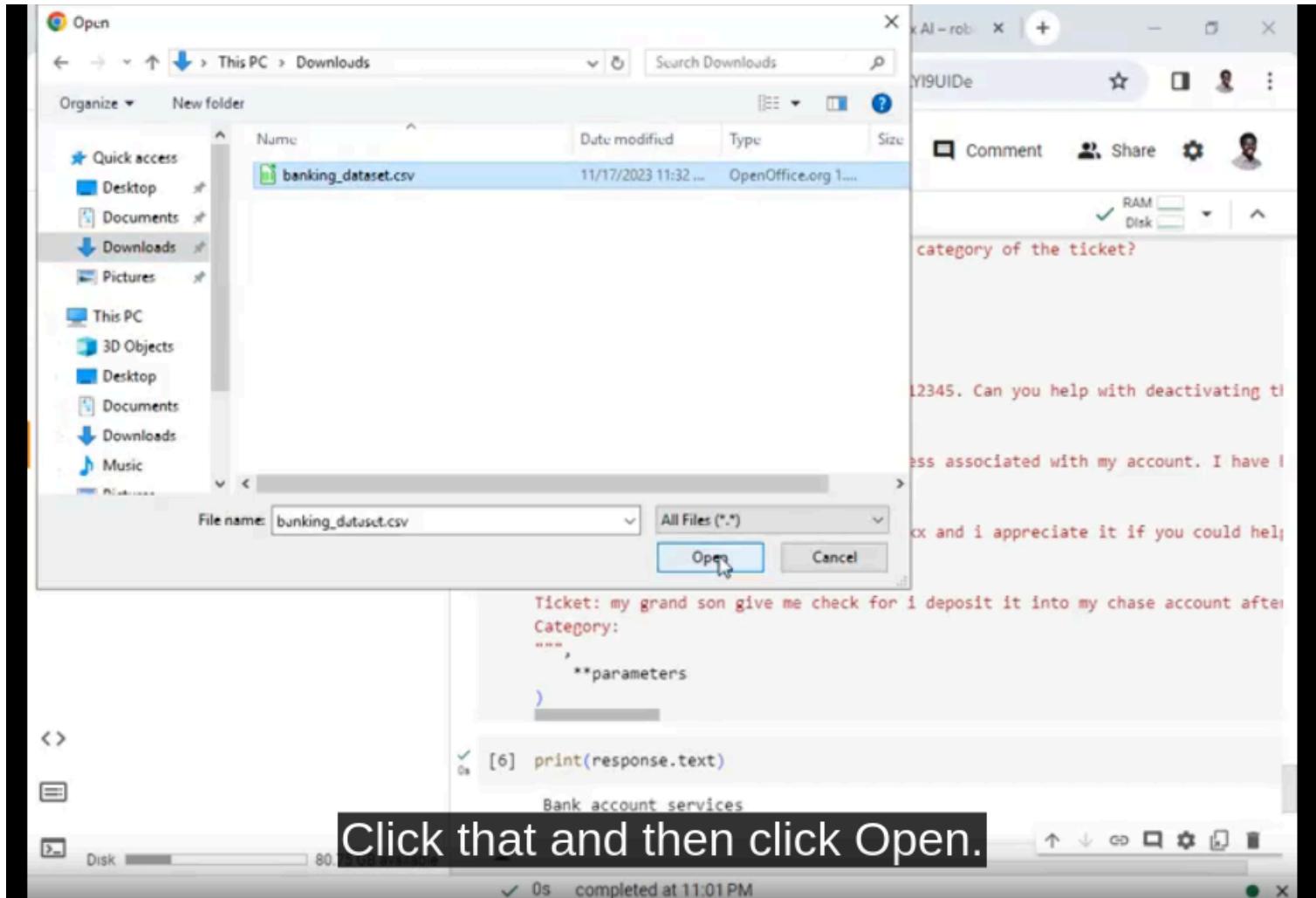
Ticket: good morning my name is xxxx xxxx and i appreciate it if you could help
Category: Loans and Mortgages

Ticket: my grand son give me check for i deposit it into my chase account after
Category:
""",
    **parameters
)

print(response.text)
```

The code editor shows the output of the predictions for each ticket example.

Update banking_dataset.csv file to colab



Load CSV file

The screenshot shows a Google Colab notebook titled "Building Vertex AI chatbot for financial institution". The code cell contains the following Python code:

```
[6] print(response.text)
Bank account services
import pandas as pd
banking_df = pd.read_csv("/content/banking_dataset.csv")
banking_df.head()
```

A large text overlay at the bottom of the screen reads: "read the top five entries in our data set."

First 5 row

The screenshot shows the same Google Colab notebook. The code cell now contains:

```
[7]
banking_df = pd.read_csv("/content/banking_dataset.csv")
banking_df.head()
```

The output area displays the first five rows of the "banking_dataset.csv" DataFrame:

	customer_id	call_date	call_time	call_transcript
0	1234567890	2023-10-31	13:55:00	Customer: Hello, I'm calling to check on the s...
1	3456789012	2023-10-31	14:05:00	Customer: Hello, I'm calling to inquire about ...
2	1098765432	2023-10-31	14:10:00	Customer: Hello, I'm calling to transfer money...
3	2134567890	2023-10-31	14:15:00	Customer: Hello, I'm calling to dispute a char...
4	4567890123	2023-10-31	14:20:00	Customer: Hello, I'm calling to ask about your...

A large text overlay at the bottom of the screen reads: "So Shift+Enter, we see these are the top five entries in our data set."

Add create predict category in dataframe by LLM

The screenshot shows a Google Colab notebook titled "Building Vertex AI chatbot for financial institution". The sidebar on the left lists files: sample_data and banking_dataset.csv. The main area displays a table of call transcripts and a code cell.

ID	Date	Time	Transcript
1	2023-10-31	14:05:00	Customer: Hello, I'm calling to transfer money...
2	2023-10-31	14:10:00	Customer: Hello, I'm calling to dispute a char...
3	2023-10-31	14:15:00	Customer: Hello, I'm calling to ask about your...
4	2023-10-31	14:20:00	Customer: Hello, I'm calling to ask about your...

In the code cell, the following Python code is shown:

```
banking_df['Classification'] = banking_df.apply(lambda x: model.predict([
    """Multi-choice problem: Define the category of the ticket?
Categories:
- Credit card
- Bank account services
- Loans and Mortgages"""
    + 'Ticket: ' + x['call_transcript'] +
    "Category:\"" + x['text'].text, axis=1))
```

A large black rectangular box covers the output of the code cell, containing the text:

So it's very,
very important to do the correct thing.

result

Generative AI | colab.google | Building Vertex | Vertex AI | Vertex AI - rob... | +

colab.research.google.com/drive/1zCi25ySsLIGFpj7VQ1JXzID4vNWGWDKR#scrollTo=orTGozdcb4v0

Building Vertex AI chatbot for financial institution

File Edit View Insert Runtime Tools Help

Files

+ Code + Text

[19] category:----).text, axis=1)

banking_df.head()

	customer_id	call_date	call_time	call_transcript	Classification
0	1234567890	2023-10-31	13:55:00	Customer: Hello, I'm calling to check on the s...	Loans and Mortgages
1	3456789012	2023-10-31	14:05:00	Customer: Hello, I'm calling to inquire about ...	Investment products
2	1098765432	2023-10-31	14:10:00	Customer: Hello, I'm calling to transfer money...	Bank account services
3	2134567890	2023-10-31	14:15:00	Customer: Hello, I'm calling to dispute a char...	Credit card
4	4567890123	2023-10-31	14:20:00	Customer: Hello, I'm calling to ask about your...	Loans and Mortgages

Shift+Enter, yeah, we can see that there's now a new column called Loans and

Disk 0s completed at 11:40 PM

Task Summary

Task Summary

 Task Objective

Learners will create a Google colab notebook cell to use the Google Cloud APIs to classify customer calls.

 Key Takeaways

- Model classification can be wrong, it is important to double check.
- Lambda can be used to apply model on all row entries in a csv at once.

Congratulations, you've gotten to the end of the fourth tasks,

Practice 2: Classify IT customer request tickets

coursera

GUIDED PROJECT

Practice 2 Activity



This task is optional and ungraded.
The goal is to check your understanding.

leveraging an IT customer request dataset and classifying that.

Practice 2 Activity

coursera

Practice 2 Activity

- Go to Google colab
- Download the dataset from the link: https://drive.google.com/file/d/11zojv07je12kZrD3tktjoM5_SOz9MOOh/view?usp=sharing can also be found in the resources section of the projects course.
- Import the data to the Google Colab
- Read the data into a variable with pandas
- Find the top 5 entries in the csv file
- Create the classification column
- Run predict model on the Ticket Description column

 **Things to Note**

→ Import the dataset into the files in the Google colab.

 **Pro Tip**

★ Need to use lambda functions for the dataframe to classify all the entries

(Note to the learner: Pause the video to complete the task and unpause to see the solution once the task is complete.)

Load CSV file

The screenshot shows a Google Colab notebook titled "Building Vertex AI chatbot for financial institution". The code cell contains the following Python code:

```
it_customer_df = pd.read_csv("/content/customertickets.csv")
it_customer_df.head()
```

The output of the code is a Pandas DataFrame named `it_customer_df`, which has the following structure:

Customer Age	Customer Gender	Product Purchased	Date of Purchase	Ticket Type	Ticket Subject	Ticket Description
32	Other	GoPro Hero	2021-03-22	Technical issue	Product setup	I'm having an issue with the {product_purchase...}
42	Female	LG Smart TV	2021-05-22	Technical issue	Peripheral compatibility	I'm having an issue with the {product_purchase...}
48	Other	Dell XPS	2020-07-14	Technical issue	Network problem	I'm facing a problem with my {product_purchase...}
27	Female	Microsoft Office	2020-11-13	Billing inquiry	Account access	I'm having an issue with the {product_purchase...}
67	Female	Autodesk AutoCAD	2020-02-04	Billing inquiry	Data loss	I'm having an issue with the {product_purchase...}

Classification label

The screenshot shows a Google Colab interface with the following details:

- Header:** colab.research.google.com/drive/1zCi25ySsLIGFpj7VQ1JXzID4vNWGWDKR#scrollTo=xRRCxo3NdPNB
- Title:** Building Vertex AI chatbot for financial institution
- File Menu:** File, Edit, View, Insert, Runtime, Tools, Help, All changes saved
- Comment and Share buttons:** Comment, Share, Settings, User profile
- RAM Disk:** RAM Disk icon with a green checkmark.
- Files Sidebar:** Shows sample_data, banking_dataset.csv, and customertickets.csv.
- Data Preview:** A table with columns: Age, Gender, Device, Purchase Date, Classification, and Description. The last column contains truncated text from the dataset.
- Code Cell:** It_customer_df['Classification'] = it_customer_df.head(100).apply(lambda x:model.
'Ticket: '+ x['Ticket Description'] +
"Category:'").text, axis=1)
- Output Cell:** [25] it_customer_df.head()
- Disk Status:** Disk 80.74 GB available

Result

```
+ Code + Text
[24]: 'Ticket: ' + x[['Ticket Description']] +
      "Category:'").text, axis=1)
```

it_customer_df.head()

Product Purchased	Date of Purchase	Ticket Type	Ticket Subject	Ticket Description	Classification
GoPro Hero	2021-03-22	Technical issue	Product setup	I'm having an issue with the {product_purchase...}	Account & Billing
LG Smart TV	2021-05-22	Technical issue	Peripheral compatibility	I'm having an issue with the {product_purchase...}	Product Defect \n\nSub-Category: Intermittent...
Dell XPS	2020-07-14	Technical issue	Network problem	I'm facing a problem with my {product_purchase...}	Product Defect
Microsoft Office	2020-11-13	Billing inquiry	Account access	I'm having an issue with the {product_purchase...}	Product Support
Autodesk AutoCAD	2020-02-04	Billing inquiry	Data loss	I'm having an issue with the {product_purchase...}	Product Quality

Practice 2 Activity

- Go to Google colab
- Download the dataset from the link: https://drive.google.com/file/d/11z0jv07je12kZrD3tktjoM5_SQz9MOOh/view?usp=sharing can also be found in the resources section of the projects course.
- Import the data to the Google Colab
- Read the data into a variable with pandas
- Find the top 5 entries in the csv file
- Create the classification column
- Run predict model on the Ticket Description column for top 100 entries



Things to Note

- Import the dataset into the files in the Google colab.

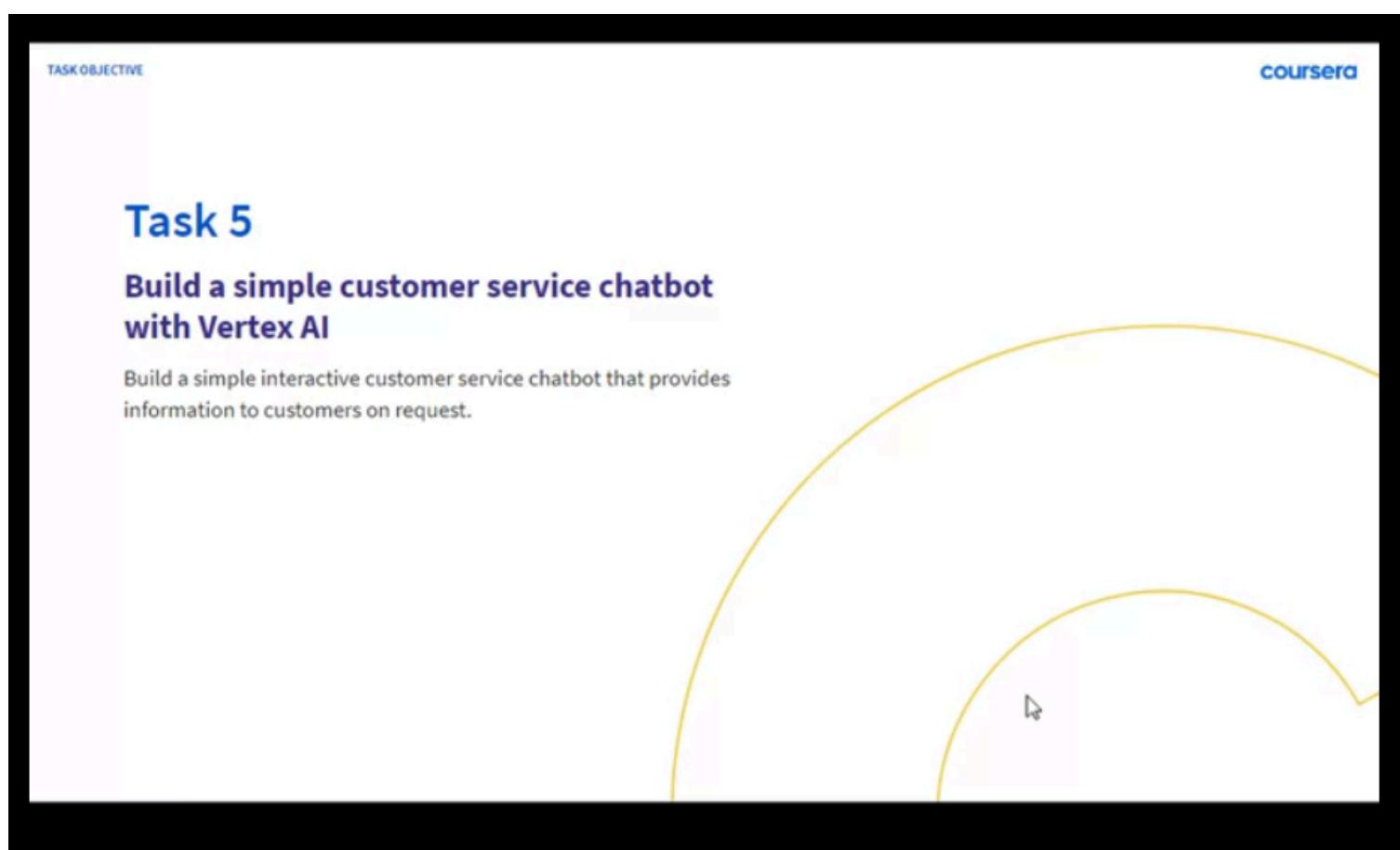


Pro Tip

- ★ Need to use lambda functions for the dataframe to classify all the entries

(Note to the learner: Pause the video to complete the task and unpause to see the solution once the task is complete.)

Task 5: Build a simple customer service chatbot with Vertex AI



Go to vertex AI → prompt example —> Customer service chatbot

A screenshot of the Google Cloud Vertex AI Language interface. The left sidebar shows "Vertex AI" selected under "TOOLS". The main panel has "Language" selected under "GET STARTED". The "Prompt examples" section shows two options: "Science chatbot" and "Customer service chatbot". The "Customer service chatbot" card is highlighted with a yellow border and has an "OPEN" button at the bottom. The URL in the browser bar is "console.cloud.google.com/vertex-ai/generative/language/gallery?project=robomuaapp".

Customer

console.cloud.google.com/vertex-ai/generative/language/prompt-examples/Customer%20service%20chat...

Google Cloud robоМУAapp Search (/) for resources, docs, prod... Search

Vertex AI Customer service chatbot SAVE VIEW CODE

Context (Optional)

We want your feedback.

Responses

USER I want to return my order. What's your return policy?

AI We offer free returns and exchanges within 30 days of your delivery, with [instructions](#)

Region us-central1 (Iowa)

Model chat-bison (latest)

Temperature 0.2

Advanced

RESET PARAMETERS

REPORT INAPPROPRIATE RESPONSES

Content processed through Vertex AI is assessed against a list of safety attributes. Confidence scores for these attributes are returned in API responses [but are not visible in the UI](#).

Dashboard Model Garden Pipelines

Colab Enterprise Workbench

Overview Language Vision Marketplace

Language

This screenshot shows the Google Cloud Vertex AI Generative AI Studio interface. On the left, there's a sidebar with sections for TOOLS (Dashboard, Model Garden, Pipelines), NOTEBOOKS (Colab Enterprise, Workbench), and GENERATIVE AI STUDIO (Overview, Language, Vision, Marketplace). The 'Language' section is currently selected. The main area is titled 'Customer service chatbot'. It has three main panels: 'Context (Optional)' containing a text input field with placeholder text; 'Examples (Optional)' containing two examples of user and AI interactions; and 'Responses' which shows a user query about returning an order and an AI response offering free returns and exchanges. To the right of these are configuration options for 'Region' (set to us-central1 (Iowa)), 'Model' (set to chat-bison (latest)), and 'Temperature' (set to 0.2). There are also links for 'Advanced' settings and 'RESET PARAMETERS'. At the bottom, there's a section for reporting inappropriate responses with a note about confidence scores.

Context “you’re a customer service representative of a financial institution dedicated to keeping customer happy”

The screenshot shows the Google Cloud Vertex AI Generative Language interface. On the left, there's a sidebar with various icons. The main area is titled "Customer service chatbot". On the left side of the main area, there are two sections: "Context (Optional)" and "Examples (Optional)". The "Context" section contains the context provided in the question: "you're a customer service representative of a financial institution dedicated to keeping customer happy". The "Examples" section contains a user message: "I was in a car accident last month and couldn't return the item in the 30 days time window. Can you make an exception?" To the right of these sections is a "Responses" pane showing an AI-generated response: "We offer free returns and exchanges within 30 days of your delivery, with exceptions as described in our Returns Policy. Certain items are designated as final sale and not eligible for returns or exchanges. All on-sale purchases are final." Below the responses is a "Continue the conversation" button with a right-pointing arrow and a microphone icon. On the far right, there are settings for the AI call: "Region" set to "us-central1 (Iowa)", "Model" set to "chat-bison (latest)", "Temperature" set to 0.2, and a "RESET PARAMETERS" button. There's also a link to "REPORT INAPPROPRIATE RESPONSES" and a note about safety attribute assessment.

Type : "Hello"

The screenshot shows the Google Cloud Vertex AI interface for creating a chatbot. The left sidebar has a 'Customer service chatbot' section selected. The main area displays a conversation window with the following messages:

- [customer name]. What is the charge that you are disputing?
- AI
- USER hello
- AI Hello, how can I help you today?

On the right side, there are configuration options:

- SAVE < VIEW CODE
- We want your feedback.
- Region: us-central1 (Iowa)
- Model: chat-bison (latest)
- Temperature: 0.2
- RESET PARAMETERS
- REPORT INAPPROPRIATE RESPONSES
- Content processed through Vertex AI is assessed against a list of safety attributes. Confidence scores for these attributes are returned in API responses but are not visible in the UI.

A note at the bottom states: Model may display inaccurate or offensive information that doesn't represent Google's view. Not all languages are supported. [Learn more](#).

In Python Colab

The screenshot shows a Google Cloud Vertex AI interface. On the left, there's a sidebar with various icons and a 'Customers' section. The main area has a title 'View code' and a sub-section 'Use this script to request a model response with a Colaboratory notebook.' It provides instructions and sample Python code for interacting with a 'ChatModel'.

```
1. Install the Vertex AI SDK for Python. After installing you'll be prompted to restart your Colab notebook runtime.  
!pip install "shapely<2.0.0"  
!pip install google-cloud-aiplatform --upgrade  
  
2. Use the following code in your notebook to request a model response  
  
from google.colab import auth as google_auth  
google_auth.authenticate_user()  
  
import vertexai  
from vertexai.language_models import ChatModel, InputOutputTextPair  
  
vertexai.init(project="robomuaapp", location="us-central1")  
chat_model = ChatModel.from_pretrained("chat-bison")  
parameters = {  
    "candidate_count": 1,  
    "max_output_tokens": 1024,  
    "temperature": 0.2,  
    "top_p": 0.8,  
    "top_k": 40  
}  
chat = chat_model.start_chat(  
    context="""you're a customer service representative of a financial institution dedicated to keeping  
examples=[  
    InputOutputTextPair(  
        input_text="""I forgot to return the item within 30 days. Can you make an exception for me?""",  
        output_text="""I understand that you want to return the item, but we are unable to return y...  
    ]  
)
```

Paste

The screenshot shows a Google Colab notebook titled "Building Vertex AI chatbot for financial institution". The left sidebar displays files: sample_data, banking_dataset.csv, and customertickets.csv. The main area contains Python code for initializing a chat model and starting a conversation:

```
from google.colab import auth as google_auth
google_auth.authenticate_user()

import vertexai
from vertexai.language_models import ChatModel, InputOutputTextPair

vertexai.init(project="robomuaapp", location="us-central1")
chat_model = ChatModel.from_pretrained("chat-bison")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 1024,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
chat = chat_model.start_chat(
    context="""you're a customer service representative of a financial institut""",
    examples=[
        InputOutputTextPair(
            input_text="""I forgot to return the item within 30 days. Can you make""",
            output_text="""I understand that you want to return the item, but we"""),
    ]
)
response = chat.send_message("""I want to return my order. What's your return po
print(f"Response from Model: {response.text}")
```

Call Model start_chat

The screenshot shows a Google Colab notebook titled "Building Vertex AI chatbot for financial institution". The code cell contains Python code for initializing a chat model and sending messages. The code imports `google.colab` for authentication, `vertexai` for the API, and `ChatModel` for the model itself. It initializes the project and location, sets parameters for the chat model, starts the chat, and sends a message about returning an item. The code is annotated with comments explaining the steps.

```
from google.colab import auth as google_auth
google_auth.authenticate_user()

import vertexai
from vertexai.language_models import ChatModel, InputOutputTextPair

vertexai.init(project="robomuaapp", location="us-central1")
chat_model = ChatModel.from_pretrained("chat-bison")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 1024,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
chat = chat_model.start_chat(
    context="you're a customer service representative of a financial institution"
)
examples = [
    InputOutputTextPair(
        input_text="I forgot to return the item within 30 days. Can you make it easier to return?",
        output_text="I understand that you want to return the item, but we"
    )
]
response = chat.send_message("""I want to return my order. What's your return policy?""")
print(f"Response from Model: {response.text}")
```

The screenshot shows the continuation of the Google Colab notebook. The code adds more examples to the chat and continues the conversation. It sends messages about returning an order, ordering 30 days ago, disputing a charge, and asking about a credit card issue. The code then asks about an insurance policy, provides more information about it, and finally asks about a disability insurance. The code is annotated with comments explaining the steps.

```
output_text="""I understand that you want to return the item, but we can make it easier. Please provide your order number so we can assist you further."""
)
]
response = chat.send_message("""I want to return my order. What's your return policy?""")
print(f"Response from Model: {response.text}")
response = chat.send_message("""I ordered 30 days ago. Could you please help me with that?""")
print(f"Response from Model: {response.text}")
response = chat.send_message("""Customer: Hello, I'm calling to dispute a charge on my account.""", **parameters)
print(f"Response from Model: {response.text}")
response = chat.send_message("""hello""", **parameters)
print(f"Response from Model: {response.text}")
response = chat.send_message("""I am having issues with my credit card""", **parameters)
print(f"Response from Model: {response.text}")
response = chat.send_message("""Tell me about your insurance policy""", **parameters)
print(f"Response from Model: {response.text}")
response = chat.send_message("""Tell me more about your insurance policy""", **parameters)
print(f"Response from Model: {response.text}")
response = chat.send_message("""Give me more information about the disability insurance""", **parameters)
print(f"Response from Model: {response.text}")
```

Task summary

The screenshot shows a 'Task Summary' section on a Coursera page. At the top right is the Coursera logo. Below it, the title 'Task Summary' is displayed in blue. To the left of the title is a yellow circular icon containing a small mountain and flag icon. The main content area has two sections: 'Task Objective' and 'Key Takeaways'. 'Task Objective' includes a description: 'Learners will build a customer service chatbot with the template from Google Vertex AI console.' To the right of this section is a small blue circular icon with a white downward arrow. 'Key Takeaways' lists one bullet point: 'The model used is chat-bison instead of text-bison for classification and summarization.'

Task Summary

 **Task Objective**

Learners will build a customer service chatbot with the template from Google Vertex AI console.

 **Key Takeaways**

- The model used is chat-bison instead of text-bison for classification and summarization.

Cumulative Task

GUIDED PROJECT

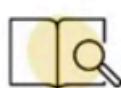
Cumulative Activity



This activity is optional and ungraded.
The goal is for you to apply the knowledge and skills learned
within this Guided Project to boost your confidence.

Cumulative Activity

1. We import vertex ai, Chatmodel and InputOutputTextPair
2. We initialize vertex ai and the chat-bison pretrained model from chatmodel as chat_model and TextGenerationModel as model
3. Initialize the parameters
4. Now we can start the chat with context
5. Create a response variable and send a couple of messages
6. Send multiple messages to interact with the chatbot
7. Append all the responses to the response variable
8. Summarize the responses.
9. Classify the responses.



Things to Note

- You need to append the responses to the same variable in order to summarize and classify

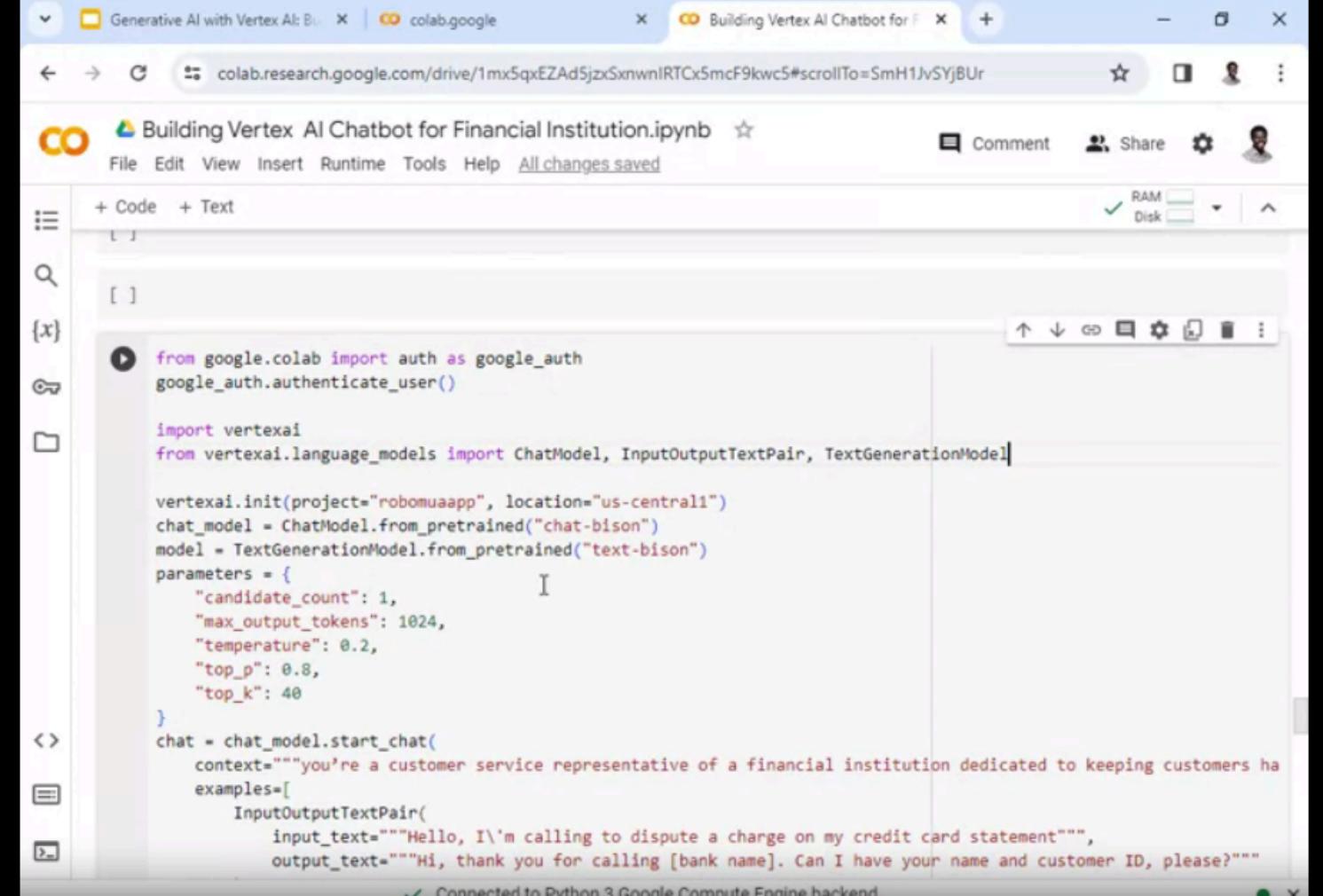


Pro Tip

- ★ Use the summarize and function calls from task 2 and 4.

(Note to the learner: Pause the video to complete the task and unpause to see the solution once the task is complete.)

Import Vertexai ChatModel , load chatmodel



```
from google.colab import auth as google_auth
google_auth.authenticate_user()

import vertexai
from vertexai.language_models import ChatModel, InputOutputTextPair, TextGenerationModel

vertexai.init(project="robomuaapp", location="us-central1")
chat_model = ChatModel.from_pretrained("chat-bison")
model = TextGenerationModel.from_pretrained("text-bison")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 1024,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
chat = chat_model.start_chat(
    context="""you're a customer service representative of a financial institution dedicated to keeping customers ha
examples=[

InputOutputTextPair(
    input_text="""Hello, I'm calling to dispute a charge on my credit card statement""",
    output_text="""Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?"""
    )]
```

Set chatbot parameters , start chat setup (context, example (input output text pair)

The screenshot shows a Google Colab notebook titled "Building Vertex AI Chatbot for Financial Institution.ipynb". The code in the cell is as follows:

```
vertexai.init(project="robomuaapp", location="us-central1")
chat_model = ChatModel.from_pretrained("chat-bison")
model = TextGenerationModel.from_pretrained("text-bison")
parameters = {
    "candidate_count": 1,
    "max_output_tokens": 1024,
    "temperature": 0.2,
    "top_p": 0.8,
    "top_k": 40
}
chat = chat_model.start_chat(
    context="""you're a customer service representative of a financial institution dedicated to keeping customers happy""",
    examples=[
        InputOutputTextPair(
            input_text="""Hello, I'm calling to dispute a charge on my credit card statement""",
            output_text="""Hi, thank you for calling [bank name]. Can I have your name and customer ID, please?"""),
        InputOutputTextPair(
            input_text="""Sure, my name is [customer name] and my customer ID is 2134567890""",
            output_text="""Thank you, Mr./Ms. [customer name]. What is the charge that you are disputing?"""),
    ]
)
responses = ''
response= chat.send_message("""Hello, I'm calling to check on the status of my loan application.""", **parameters)
print(f"Response from Model: {response.text}")
responses+=response.text
```

The status bar at the bottom indicates "Connected to Python 3 Google Compute Engine backend" and the time "1:52".

Test send Message response

The screenshot shows the same Google Colab notebook. The code is identical to the previous one, but the output cell now displays the responses from the chatbot:

```
Response from Model: Hi, thank you for calling [bank name]. Can I have your name and loan application ID, please?
Response from Model: Thank you, Mr./Ms. [customer name]. Let me check on the status of your loan application. It looks like it's still pending review.
```

The status bar at the bottom indicates "Connected to Python 3 Google Compute Engine backend" and the time "1:52".

TextGenerate model Summarize the conversion ,

The screenshot shows a Google Colab notebook titled "Building Vertex AI Chatbot for Financial Institution.ipynb". The notebook interface includes a toolbar with file operations, a sidebar with search and file navigation, and a main workspace divided into code and text sections.

In the code section, the following Python code is visible:

```
responses = ''
response= chat.send_message("""Hello, I'm calling to check on the status of my loan application.""", **parameters)
print(f"Response from Model: {response.text}")
responses+=response.text
response= chat.send_message("""Sure, my name is [customer name] and my customer ID is 1234567890""", **parameters)
print(f"Response from Model: {response.text}")
responses+=response.text

Response from Model: Hi, thank you for calling [bank name]. Can I have your name and loan application ID, please?
Response from Model: Thank you, Mr./Ms. [customer name]. Let me check on the status of your loan application. It lo
```

In the text section, the following command is shown:

```
model.predict('Summarize the following conversation between a service rep and a customer in a few sentences. Use only the most recent message in the conversation to generate the summary.')
```

Below the command, a summary of the conversation is provided:

A customer inquires about the status of their loan application. The service rep informs them that the application is still under review and that they will be notified as soon as a decision has been made.

In another cell, the following code is shown:

```
[5] model.predict("""Multi-choice problem: Define the category of the ticket?
Categories:
- Credit card
- Bank account services
- Loans and Mortgages """+ responses + """ Category:""")
```

The output of this cell is:

Loans and Mortgages

At the bottom of the notebook, a status bar indicates: "Connected to Python 3 Google Compute Engine backend".

Use TextGenerate model classificate call Categories

The screenshot shows a Google Colab notebook titled "Building Vertex AI Chatbot for Financial Institution.ipynb". The code cell contains the following Python code:

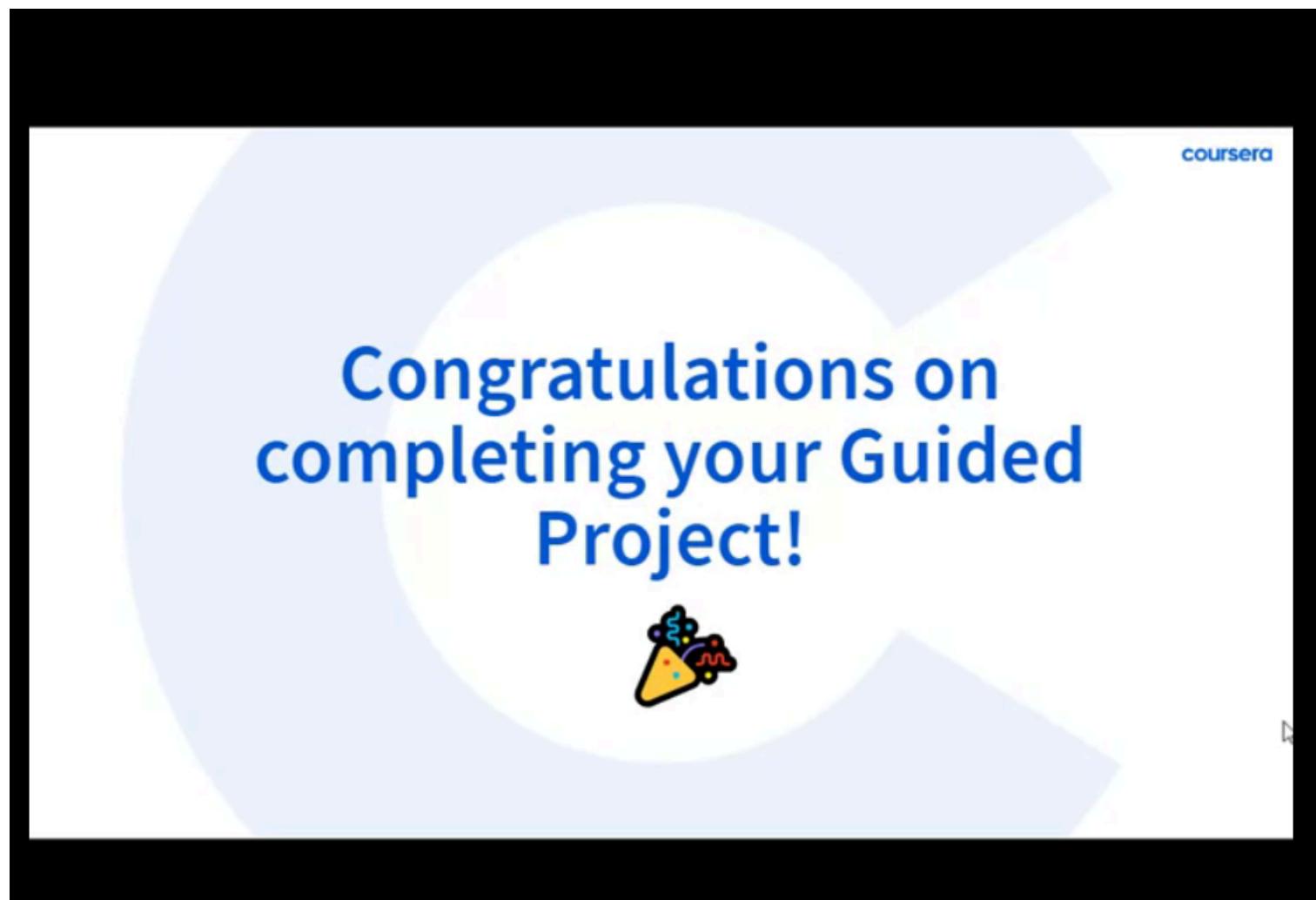
```
responses+=response.text
response= chat.send_message("""Sure, my name is [customer name] and my customer ID is 1234567890
print(f"Response from Model: {response.text}")
responses+=response.text

Response from Model: Hi, thank you for calling [bank name]. Can I have your name and loan application ID, please?
Response from Model: Thank you, Mr./Ms. [customer name]. Let me check on the status of your loan application. It lo
[3] and a customer in a few sentences. Use only the information from the conversation.' + responses, **parameters)

A customer inquires about the status of their loan application. The service rep informs them that the application
is still under review and that they will be notified as soon as a decision has been made.

model.predict("""Multi-choice problem: Define the category of the ticket?
Categories:
- Credit card
- Bank account services
- Loans and Mortgages """+ responses + """ Category:""")
Loans and Mortgage
[]
```

The code uses the `chat.send_message` method to interact with a chatbot and collect responses. It then prints the response text and concatenates it back into the `responses` variable. A descriptive comment explains the purpose of the code. Finally, it uses the `model.predict` method to classify the ticket into one of three categories: Credit card, Bank account services, or Loans and Mortgages.



Assess Your Knowledge

✓ Congratulations! You passed!

Grade received **83.33%**

Latest Submission
Grade 83.33%

To pass 80% or higher

[Go to next item](#)

1. Order the steps for Vertex AI summarization in colab.

1 / 1 point

1. Initialize VertexAI
2. Create the pre-trained model variable
3. Import VertexAI and TextGenerationModel
4. Set parameters
5. Summarizing with model

- 1,2,3,4,5
- 4,2,1,5,3
- 3,1,4,2,5
- 3,2,4,1,5

✓ Correct

Correct! Importing the necessary modules, initializing vertex AI, setting the parameters, creating the pre-trained model variable and summarizing with the model is the order of steps to take.

2. Vertex AI has an amazing portal to test out the various AI capabilities but to fully leverage all the capabilities for Vertex AI for a business it is essential to use...

1 / 1 point

- Google Cloud Console
- Google Colab
- Google Cloud Customer Service
- Google Vertex AI SDK

 **Correct**

Correct! The Vertex AI SDK allows developers to be able to fully leverage the full capabilities of Vertex AI and incorporate these capabilities in their own native applications and services.

3. In building chatbots with Vertex AI, what one essential thing is necessary to ensure that the chatbot is working relative to the unique business use case?

1 / 1 point

- Having a intro call like "Hey Google" or "Alexa"
- Setting a fixed input text at all times.
- Letting a human respond to users through the UI
- Setting the context flag.

 **Correct**

Correct! Without the context, the chatbot will just be generic and provide regular answers not necessarily related to the task required for the chatbot.

4. Select the next line of code to ensure the following code snippet works

1 / 1 point

```
import vertexai
from vertexai.language_models import ChatModel, InputOutputTextPair

vertexai.init(project="robomuaapp", location="us-central1")
chat_model = ChatModel.from_pretrained("chat-bison")
parameters = { "candidate_count": 1, "max_output_tokens": 1024, "temperature": 0.2, "top_p": 0.8,
"top_k": 40}

....
```

- response = model.predict(""" Hi, you've called Swish Bank, how may I help you?I'm a customer of your bank and I have some errors on my account details.What type of errors?""")
- chat = chat_model.start_chat(context="""you're a customer service representative of a financial institution dedicated to keeping customers happy.""")
- chat = text_model.start_chat(context="""you're a customer service representative of a financial institution dedicated to keeping customers happy.""")
- response = model.predict("""Multi-choice problem: Define the category of the ticket?Categories:
 - Credit card
 - Bank account services
 - Loans and Mortgages

Ticket: my issue is related to my credit card

Category:

""")



Correct

Correct! For the Chatmodel, starting the chat with the start_chat function and setting the context is essential.

5. To install Vertex AI in Google Colab, you need to run.....

1 / 1 point

- !pip install vertex
- install vertex-ai-google
- !pip install vertex-ai-google
- pip install google-vertex



Correct

Correct. This is the right syntax for installing a python module in the Google Colab cell.

6. PALM2 is an amazing LLM by Google that can be leveraged for all of the following **except...**

0 / 1 point

- Image generation
- Human Reasoning.
- Code generation
- Identifying toxic content



Incorrect

Incorrect. Review Task 1: Summarize customer calls in Vertex AI console