**Lab 18: Generating post-call transcription and analysis with Azure OpenAI Service**

**Introduction**

In this lab, you'll delve into the world of post-call transcription and analysis using the powerful tools provided by Azure OpenAI Service. In this lab, you'll be performing sentiment analysis and summarization using call center transcriptions. You'll be transcribing the customer recording to text, then you'll be using OpenAI to detect sentiment. Then, you'll summarize long text into a few sentences for further analysis. By the end of this exercise, you'll have gained valuable insights into leveraging Azure OpenAI for handling customer interactions efficiently.

**Objectives**

The primary objective of this lab is to equip you with the necessary skills to perform sentiment analysis and summarization on call center transcriptions using Azure OpenAI Service. To achieve this goal, you have to:

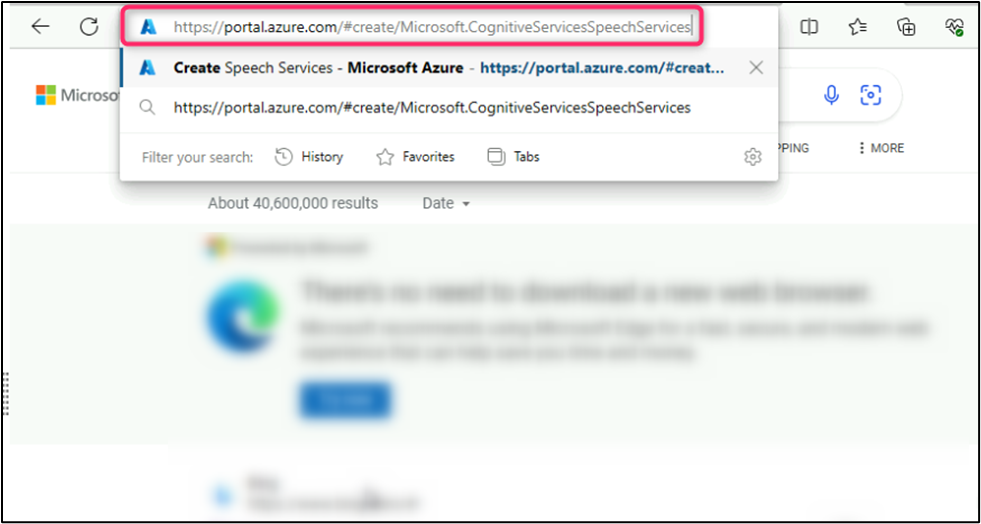
* Create a Speech resource in the Azure portal.
* Create gpt-3.5-turbo model in Azure AI Studio.
* Set environment variables to connect your resources for seamless integration.
* Execute sentiment analysis and summarization using call center transcriptions.

**Prerequisites**

* Before starting this lab, ensure to complete **Lab 01- Provisioning Azure OpenAI resource** and **Lab 09: Enabling conversation using Azure Speech Service**.

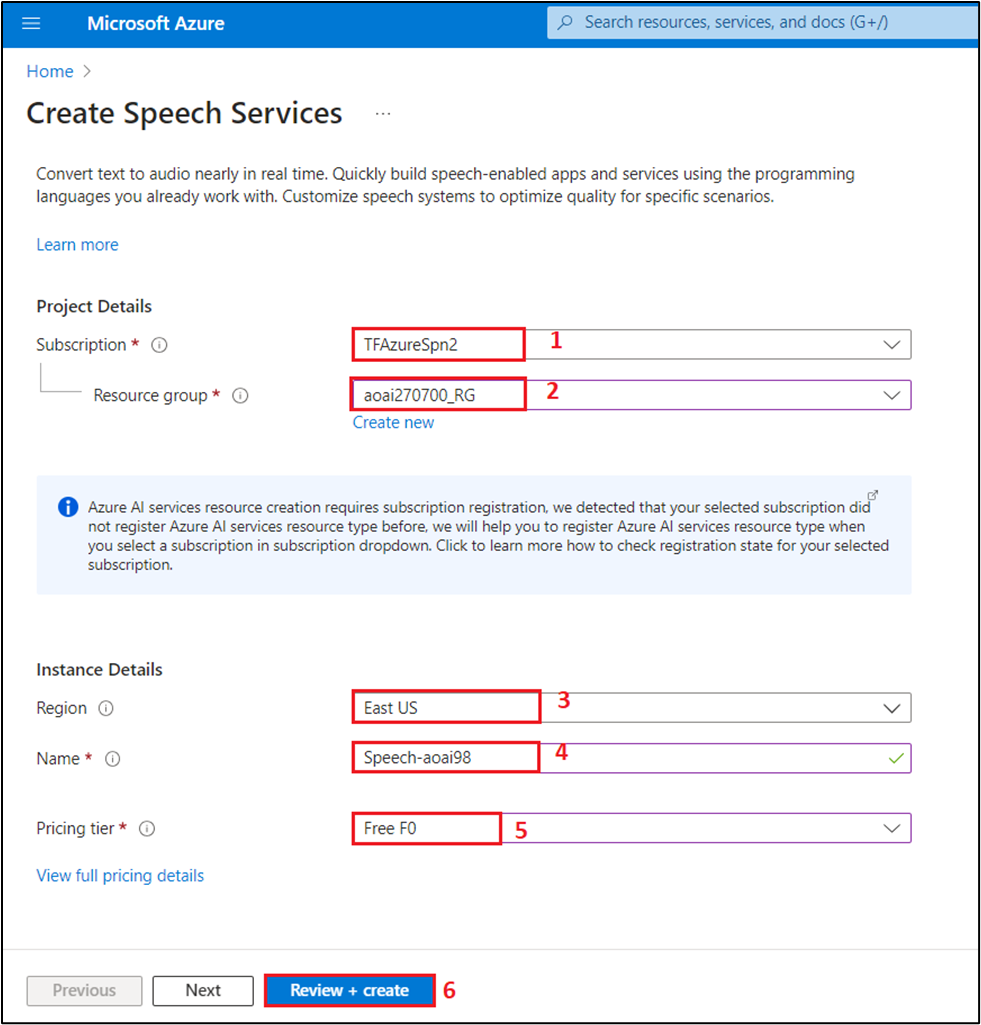
**Task 1: Create a Speech resource in the Azure portal**

1. Open your browser, in the address bar, copy and paste the following URL and press the **enter** button **https://portal.azure.com/#create/Microsoft.CognitiveServicesSpeechServices**

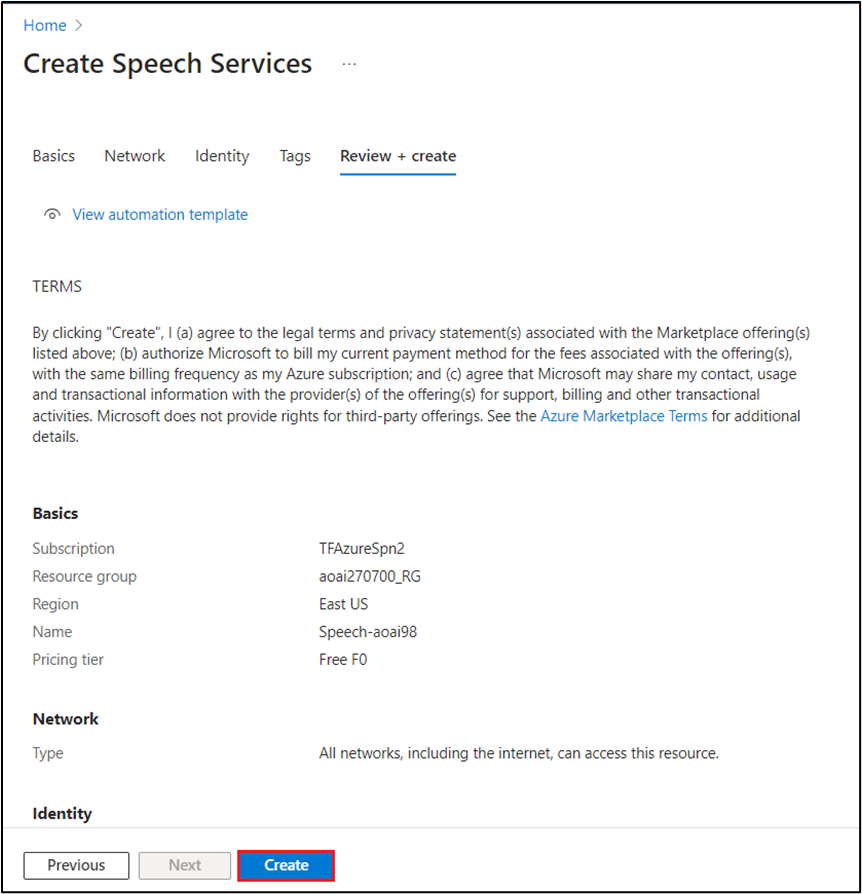


1. On the **Create Speech Services** page, provide the following information and click on **Review+create** button.

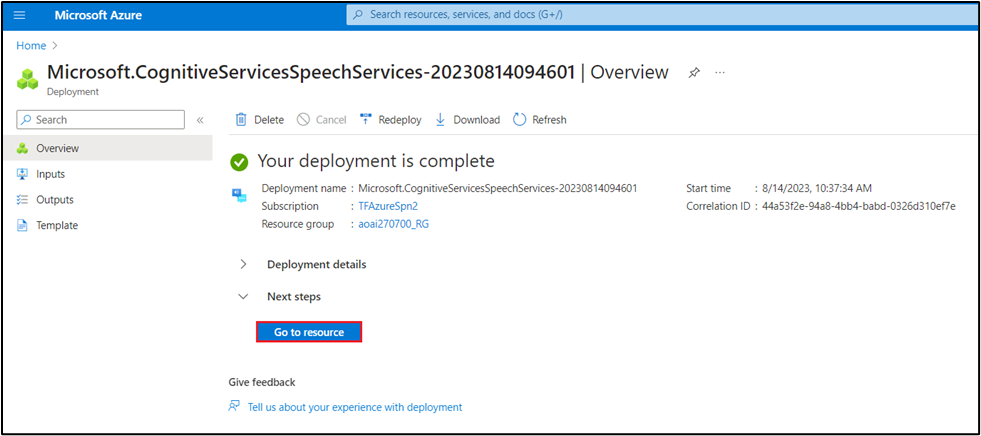
| **Field** | **Description** |
| --- | --- |
| **Subscription** | Select the subscription assigned to |
| **Resource group** | Select the **Resource Group** you had created in Lab 1 |
| **Region** | **East US** |
| **Name** | **Speech-aoaiXX** (XX can be unique number) |
| **Pricing Tier** | Free F0 |
|  |  |



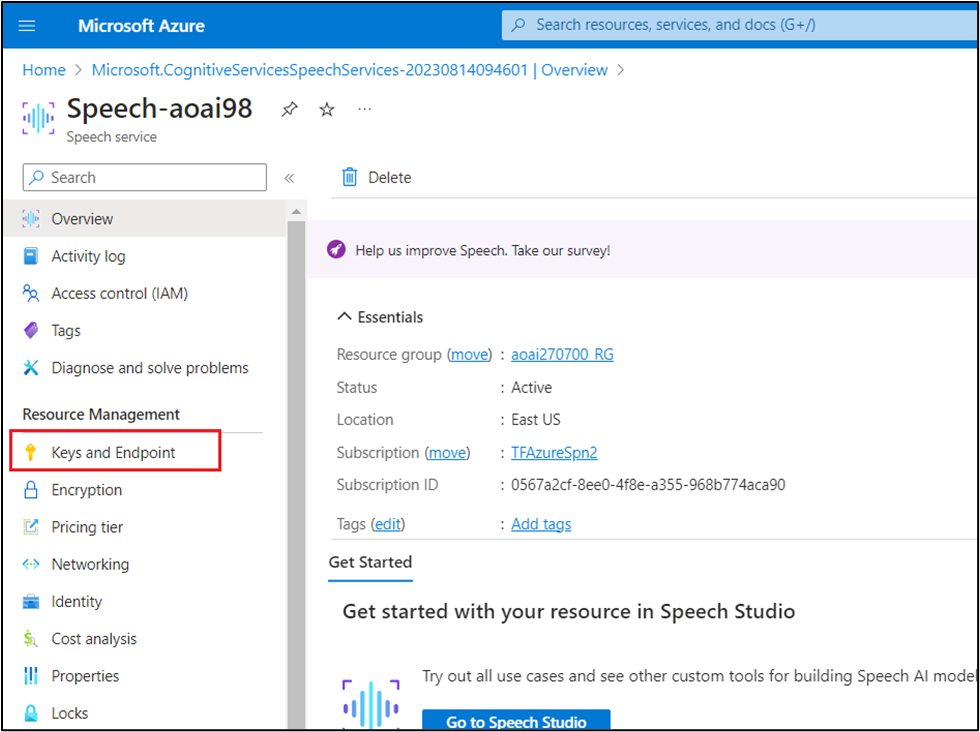
1. Once the Validation is successful, click on **Create.**



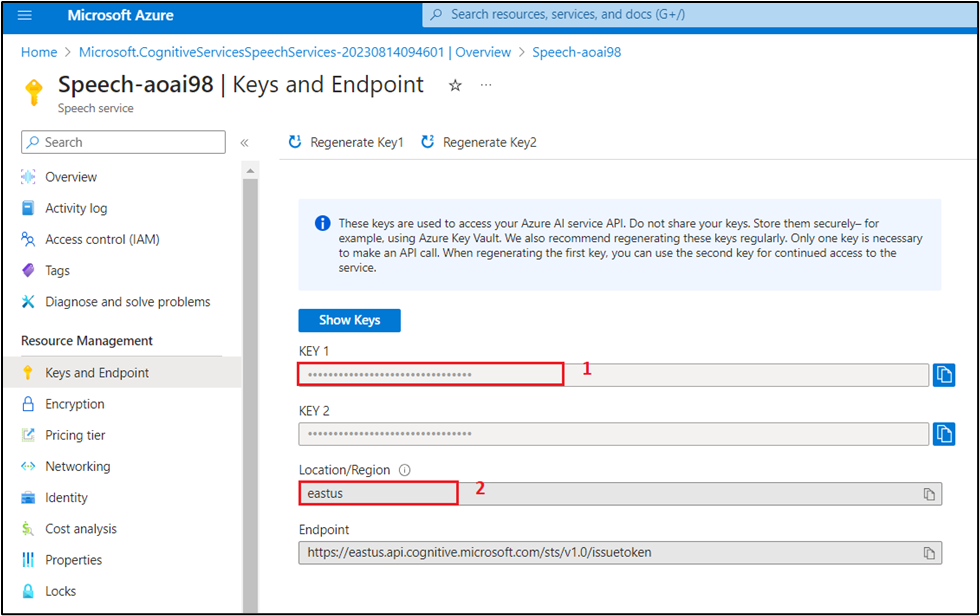
1. Once the deployment is completed, click on **Go to resource** button.

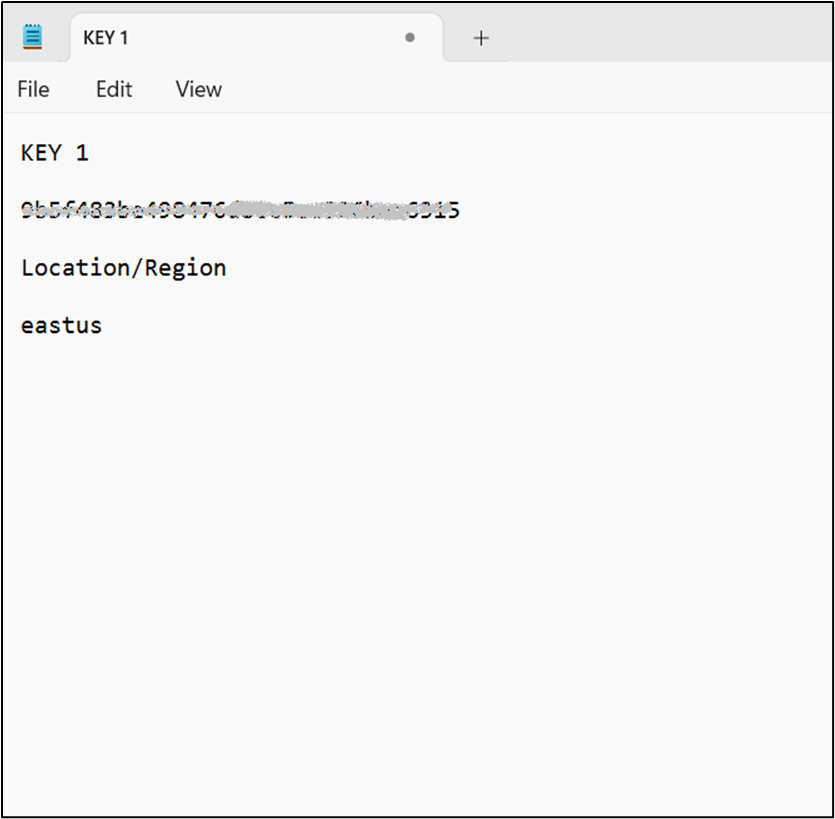


1. On **Speech-aoaiXX speech service** page, navigate to **Resource Management** and click on **Keys and Endpoints**.



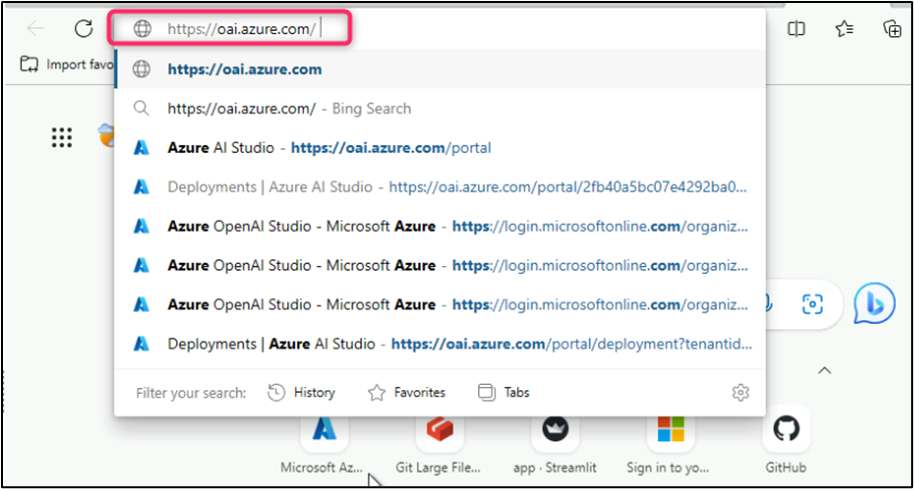
1. In **Keys and Endpoints** page, copy **KEY 1** (*You can use either KEY 1 or KEY 2)*, and **Location** and save them in a notepad as shown in the image to use them in **.env** file.



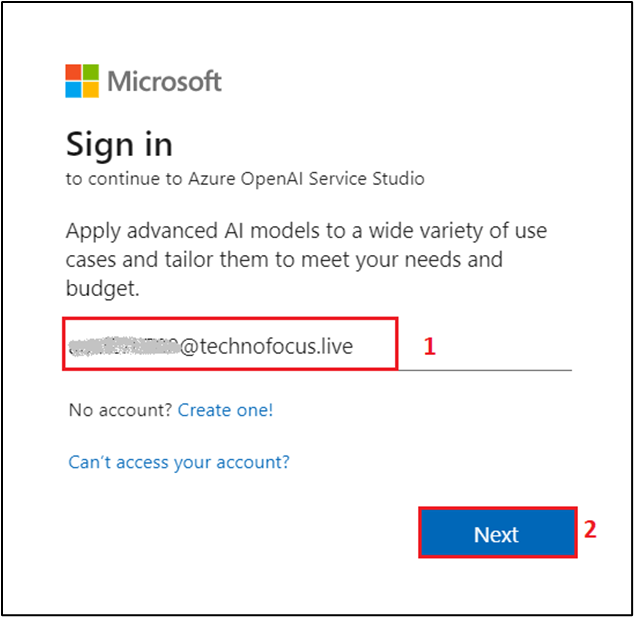


**Task 2:** Create gpt-35-turbo in Azure AI Studio

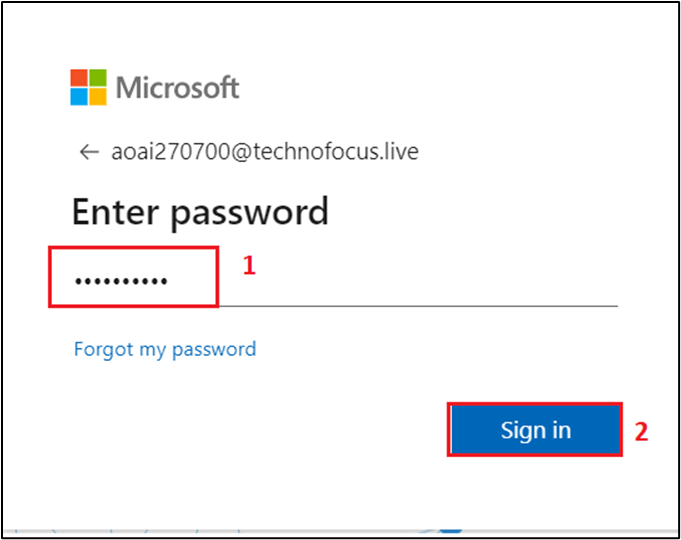
1. Open your browser, navigate to the address bar, and type or paste the following URL: **https://oai.azure.com/** then press the **Enter** button.



1. In the **Microsoft Azure** window, enter your **Sign-in** credentials, and click on the **Next** button.



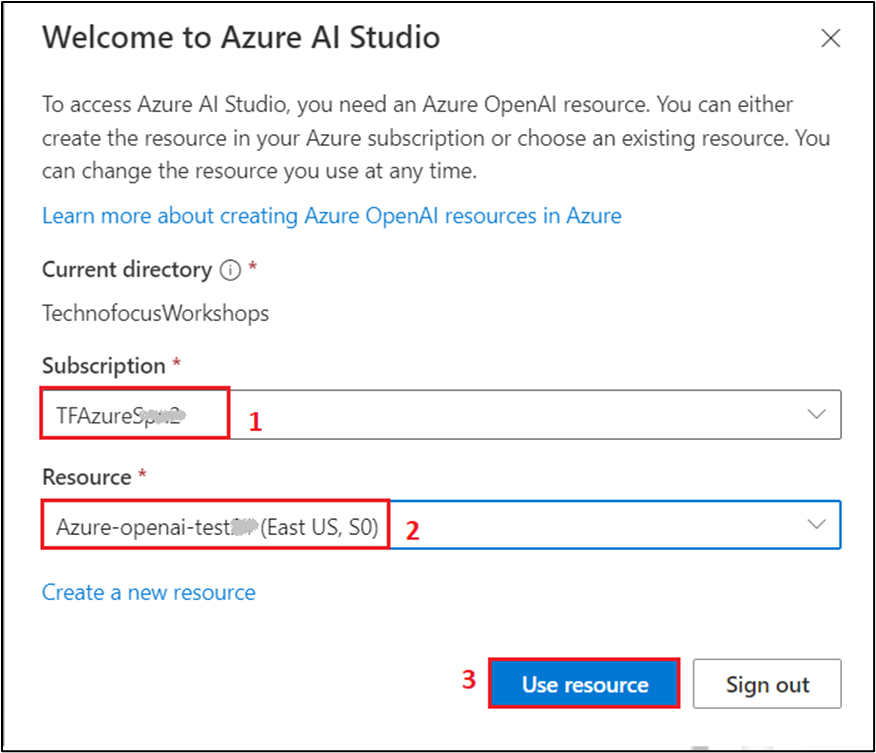
1. Then, enter the password and click on the **Sign in** button\*\*.\*\*



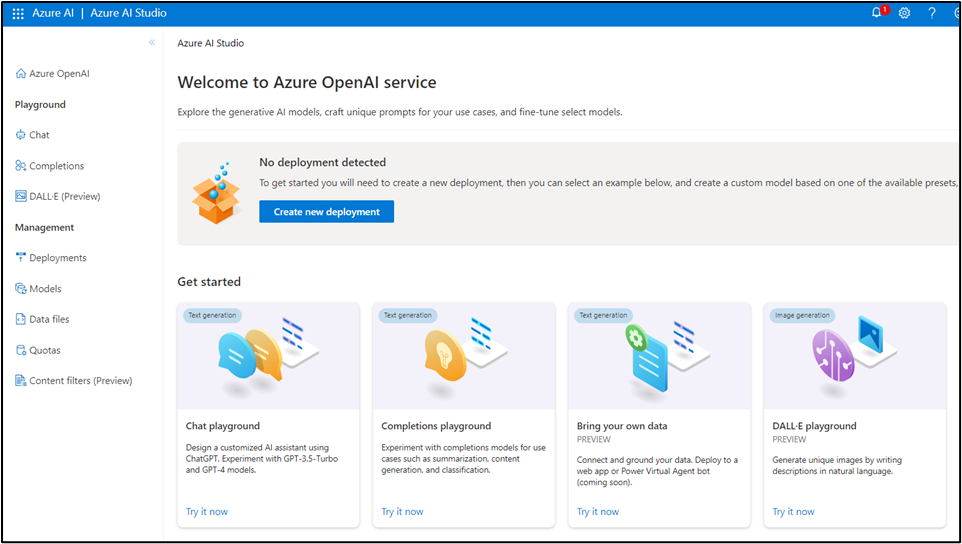
1. In **Stay signed in?** window, click on the **Yes** button.



1. On the **Welcome to Azure OpenAI Studio** dialog box, under the **Subscription** field, enter the subscription assigned to you, and in the **Resource** field, enter the assigned Resource, and click on the **Use resource** button.



1. Wait for the Azure OpenAI studio to launch.



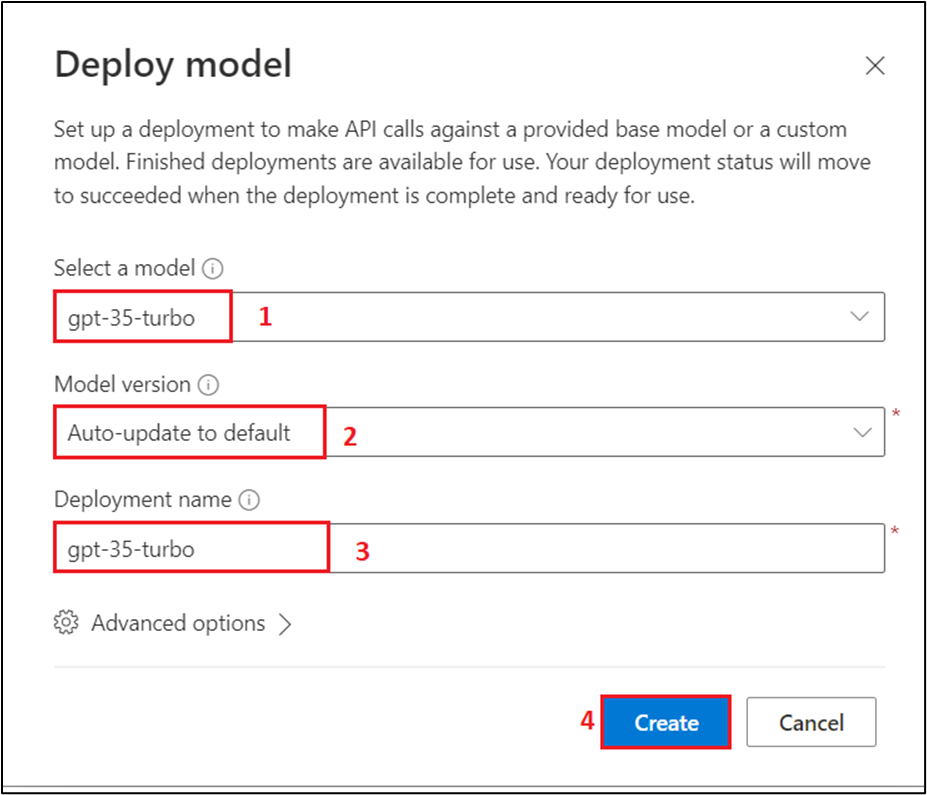
1. On the **Azure OpenAI Studio** homepage, click on **Create new deployment** button.



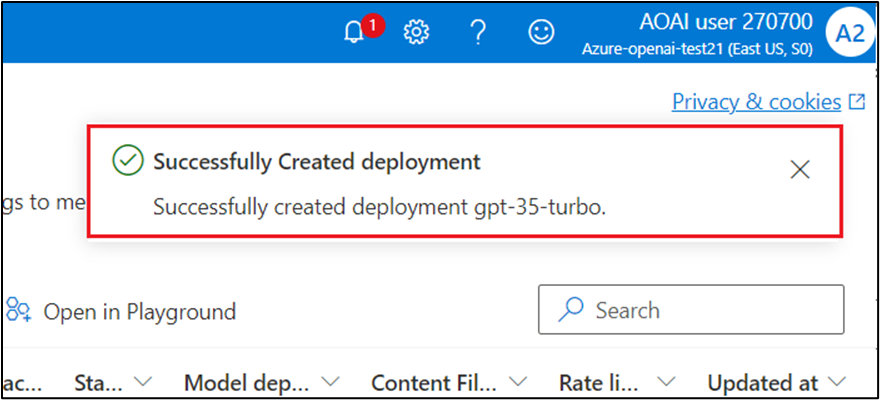
1. In the **Deployments** page, click on +**Create new deployment**.



1. In the **Deploy model** dialog box, under **Select a model**, click on the dropdown and select **gpt-35-turbo**, under **Model version**, click on the dropdown and select **Auto-update to default**, then under **Deployment name,** enter **gpt-35-turbo**. Click on the **Create** button to deploy gpt-35-turbo model.



1. You will see a notification -- **Successfully Created deployment** when the deployment is succeeded (You can also view the notification by clicking on the bell icon beside **Azure AI | Azure AI Studio**).



**Task 3: Set environment variables**

1. In your windows search bar, type **Command Prompt**. In the **Command Prompt App** dialog box, navigate and click on **Run as administrator**. If you see the dialog box - **Do you want to allow this app to make changes to your device?** then click on the **Yes** button.

A screenshot of a computer

Description automatically generated

1. Install the azure-ai-formrecognizer with the following command

pip install azure-ai-formrecognizer==3.2.

A screenshot of a computer program

Description automatically generated

1. Install the azure-cognitiveservices-speech with the following command

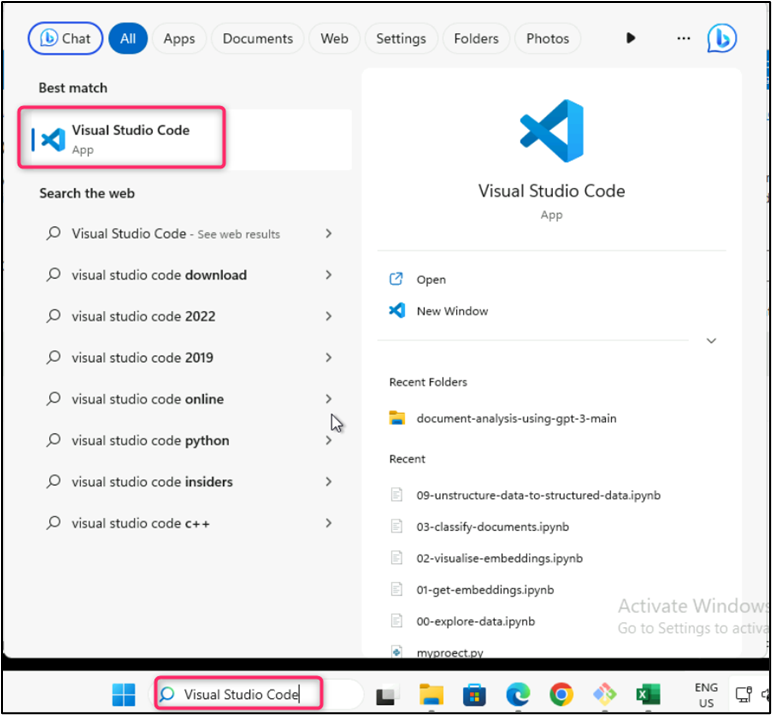
pip install azure-cognitiveservices-speech

A screenshot of a computer screen

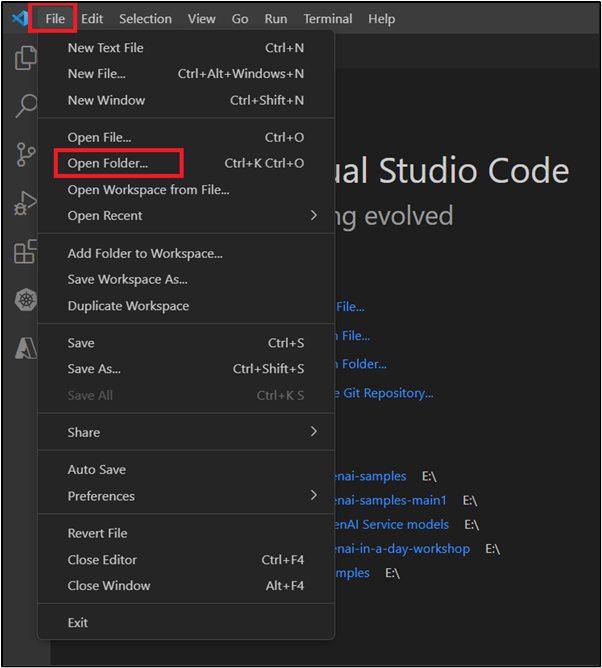
Description automatically generated

1. In your Windows search box, type Visual Studio Code, then click on **Visual Studio Code**.

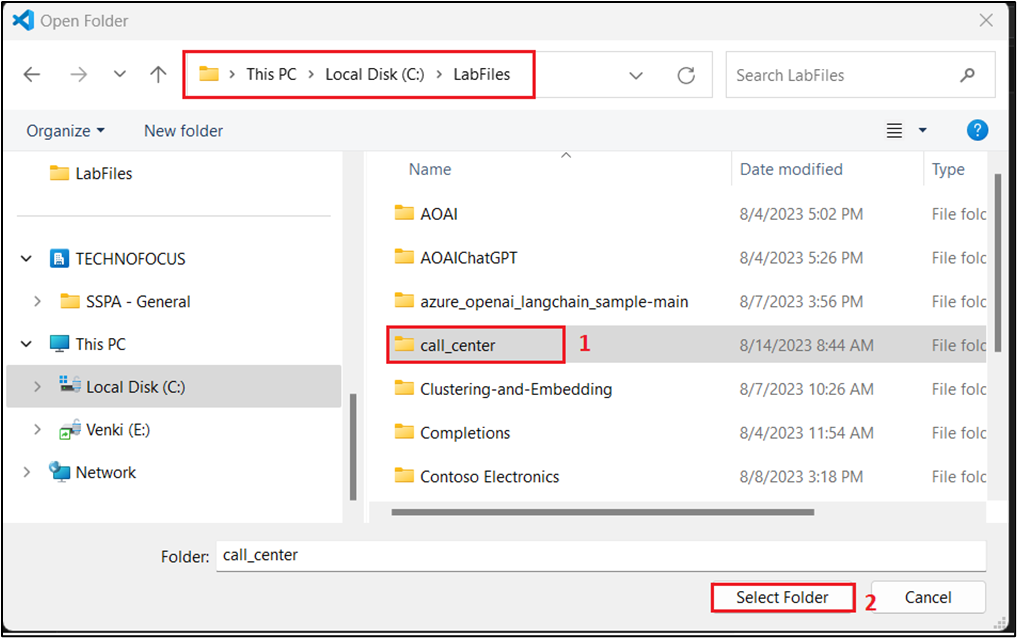
Note: Ensure that you've completed



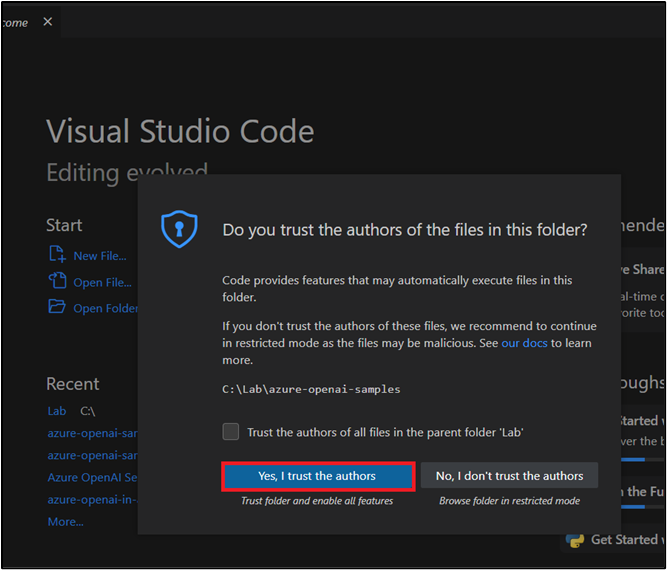
1. In the **Visual Studio Code** editor, click on **File**, then navigate and click on **Open Folder.**



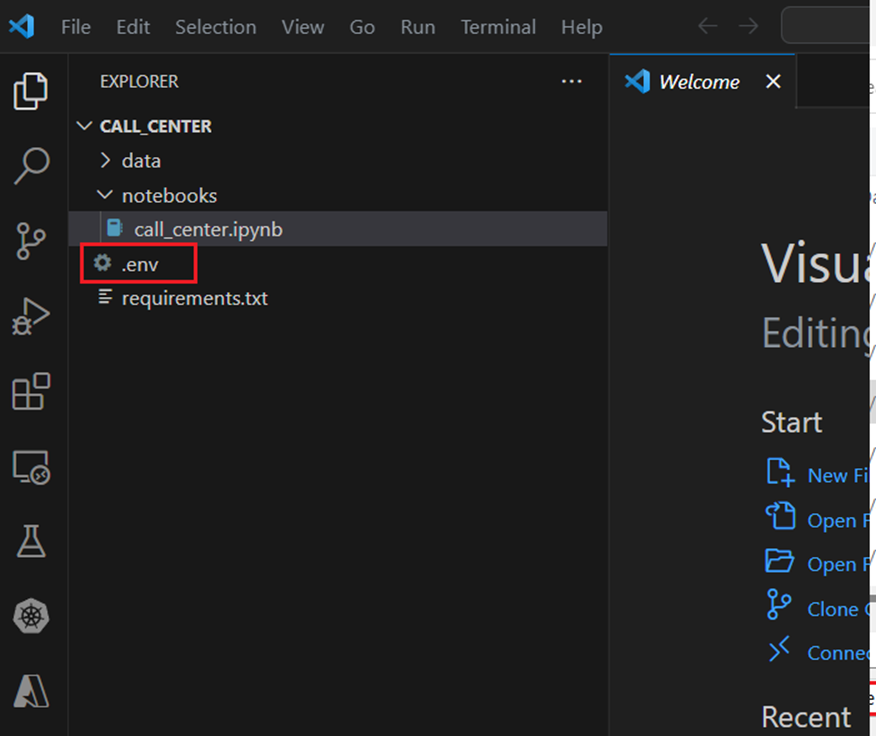
1. Navigate to **C:\LabFiles,** select **call\_center** folder.



1. If you see a dialog box - **Do you trust the authors of the files in this folder?** then click on **Yes, I trust the author**.

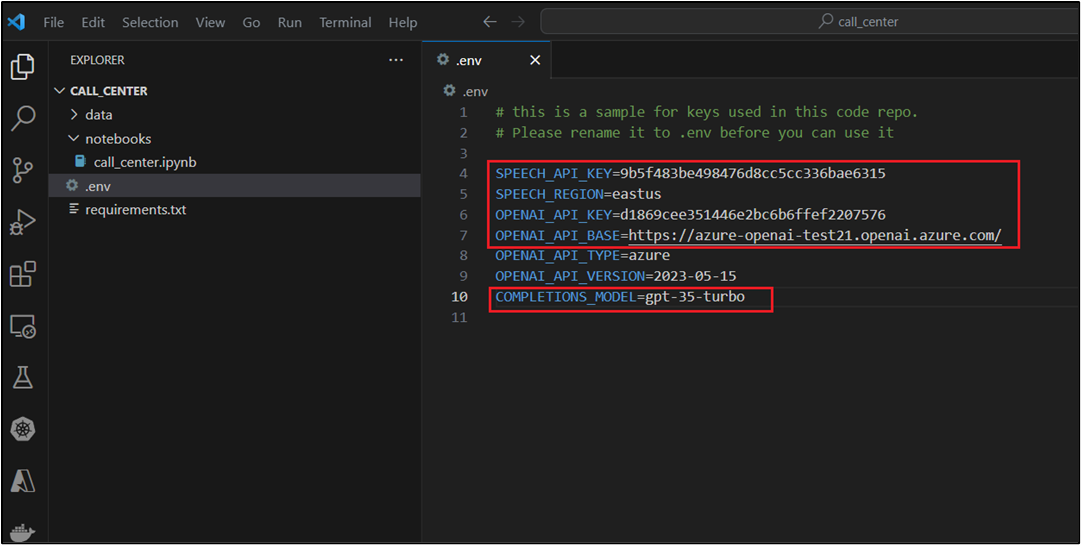


1. In Visual Studio Code, under **CALL\_CENTER,** click on **.env** file.

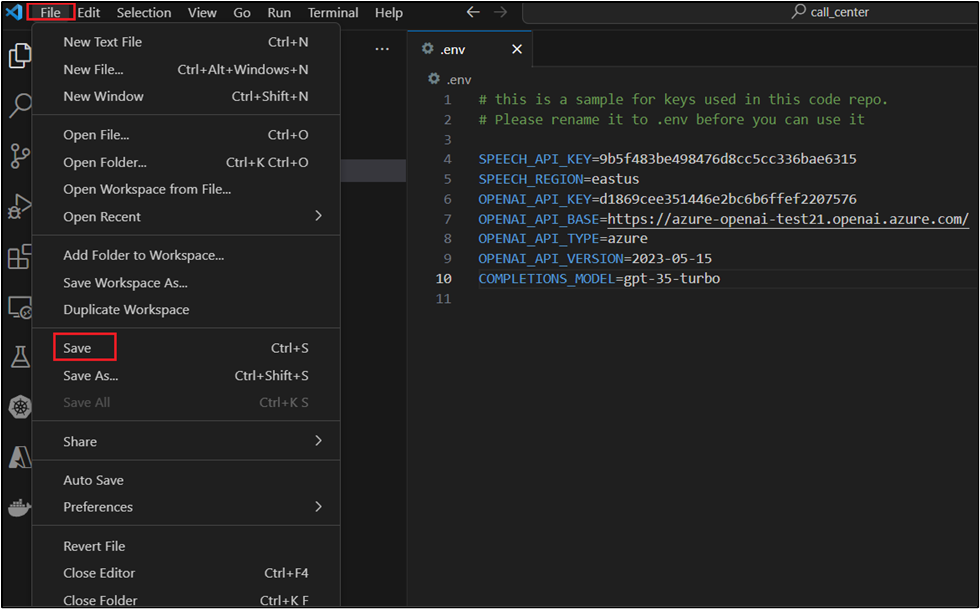


1. In the **.env** file, set the environment variables of Speech service **Key** and **Region** that you have saved in your notepad in the previous task.
2. Set the environment variables of Azure OpenAI **Key** and **Endpoint** that you have saved in your notepad in the **Lab1>Task 2**.

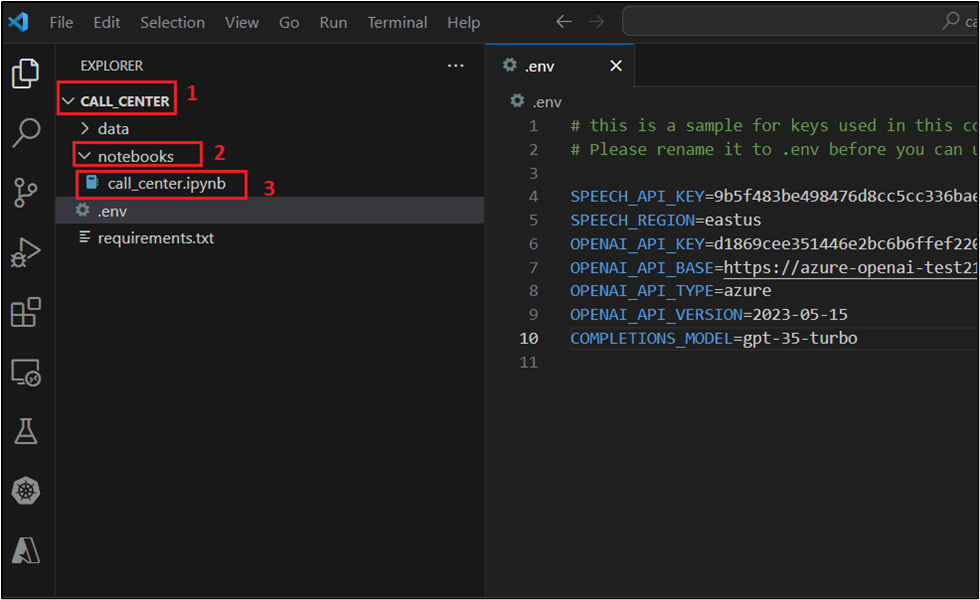
**Note**: In case, you did not find **SPEECH\_API\_KEY** and **SPEECH\_REGION** in the .env file, then insert it.



1. Save the **.env** file.



1. In Visual Studio Code, under **call\_center**, click on **notebooks**, then click on **call\_center.ipynb** notebook.

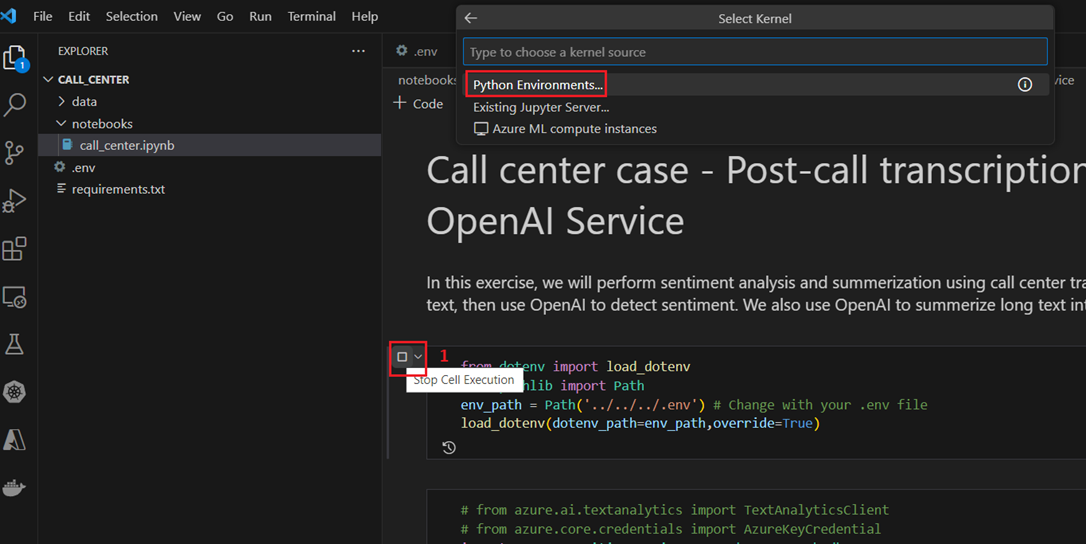


**Task 4: Sentiment analysis and summarization using call center transcriptions**

1. Ensure that the Cell Language Mode is **Python**. Click on **Execute cell start icon** as shown in the image.



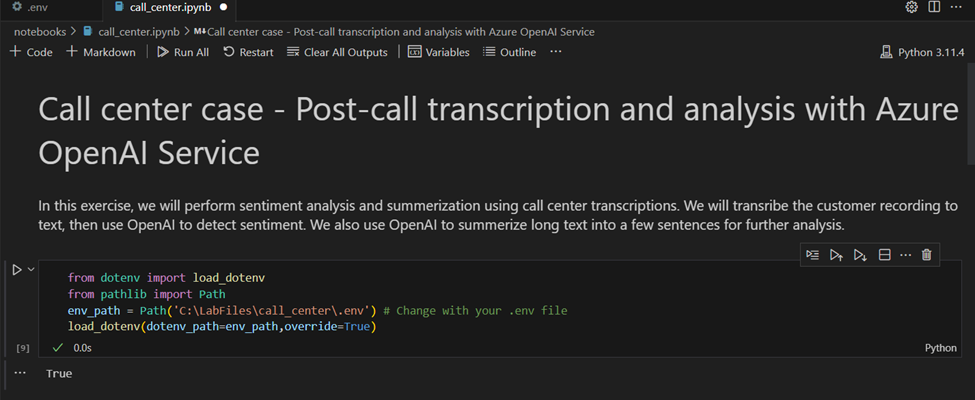
1. If prompted to select the environment, then select **Python Environments** as shown in the image.

. 

1. If prompted to select the path, then select the **Python version 3.11.5(or later version)** path as shown in the image.

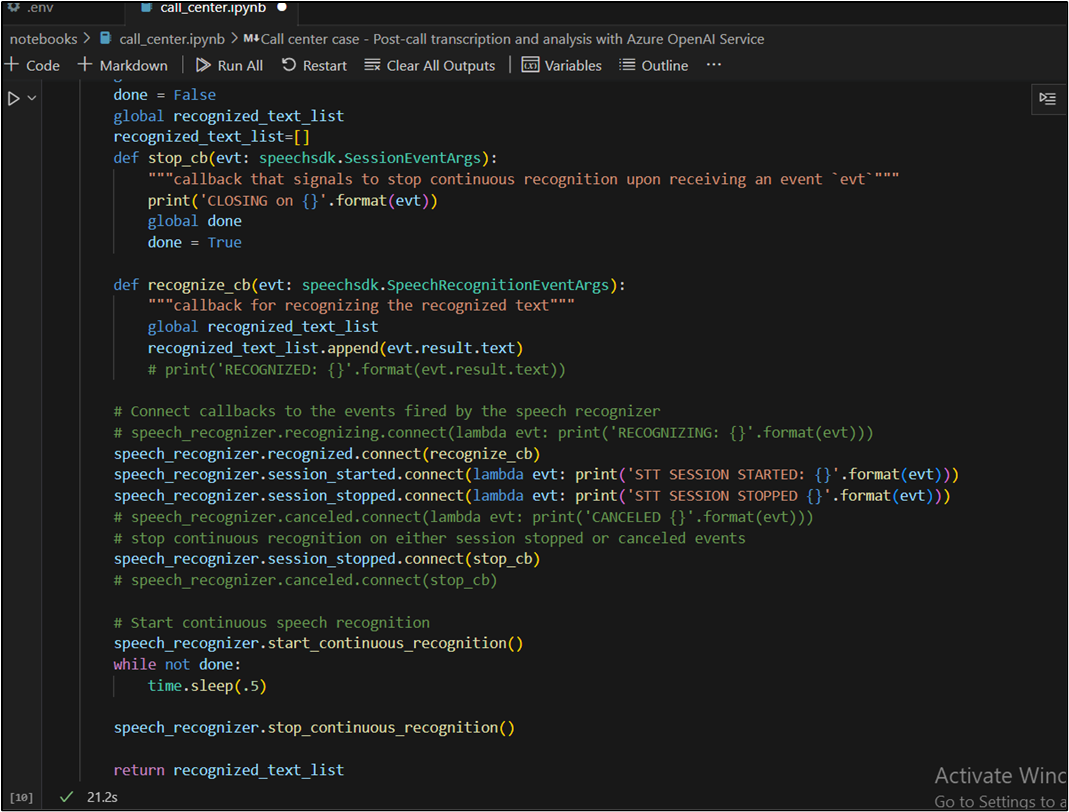
A screenshot of a computer

Description automatically generated

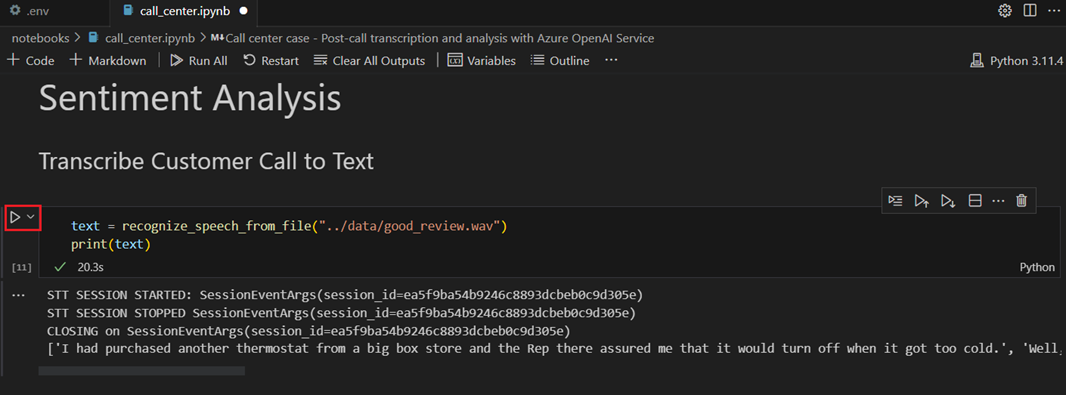


1. Similarly, execute the 2nd cell.

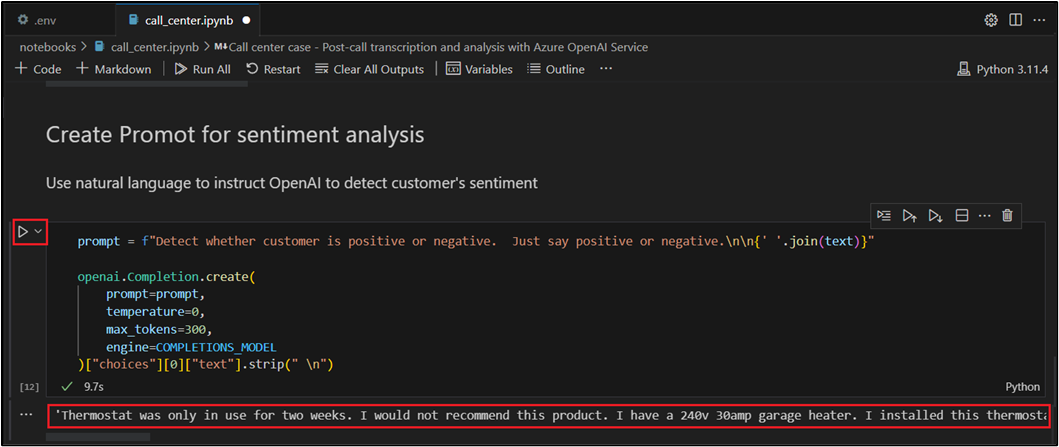




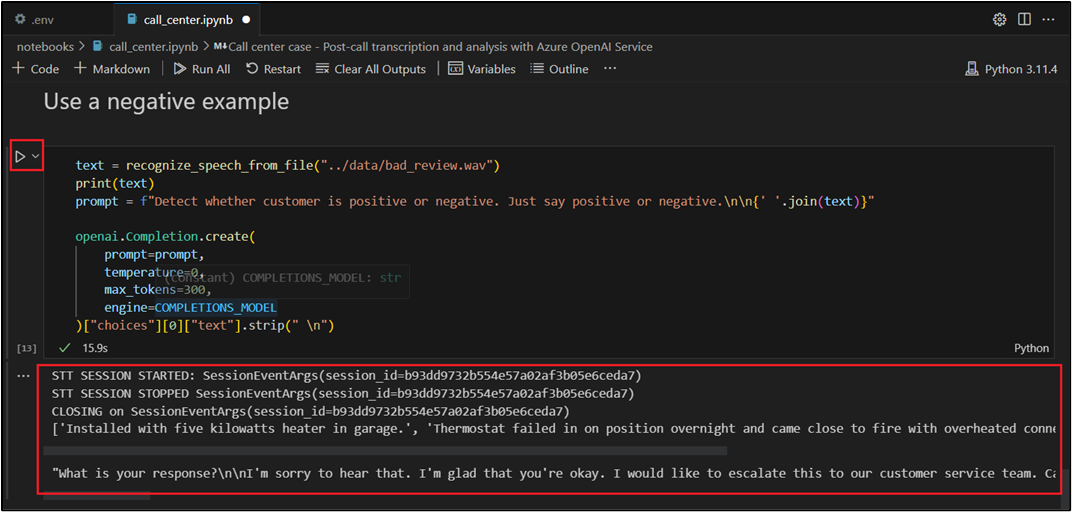
1. Under **Transcribe Customer Call to Text** heading, execute the cell by clicking on the **play** button.



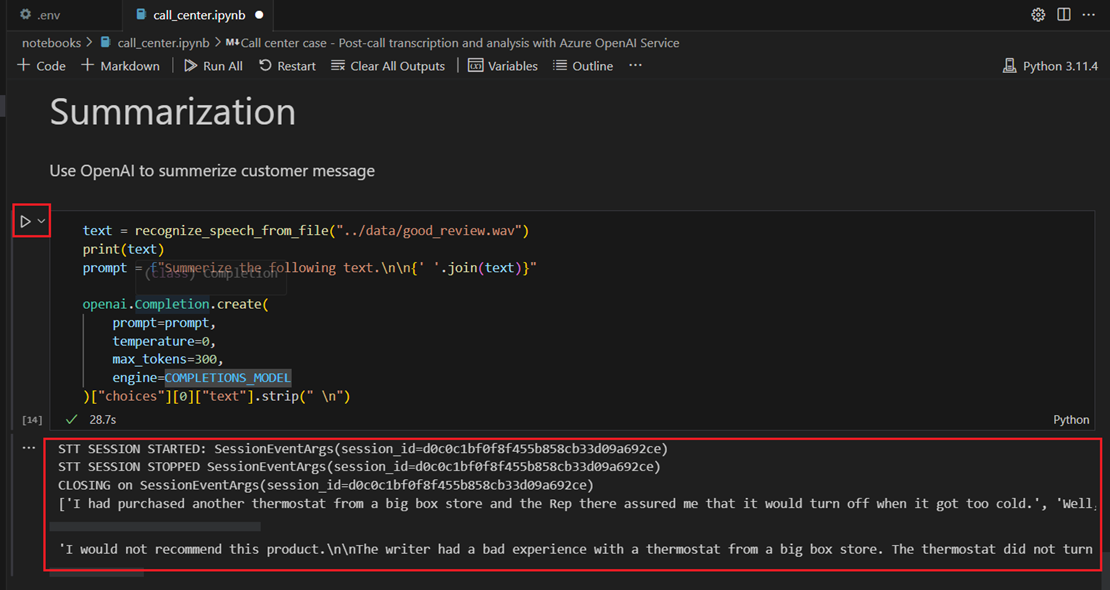
1. Under **Create Prompt for sentiment analysis** heading, execute the cell by clicking on the **play** button.



1. Under **Use a negative sample** heading, execute the cell by clicking on the **play** button.



1. Under the **Summarization** heading, execute the cell by clicking on the **play** button.



**Conclusion**

In this lab, you have embarked on a journey to harness the capabilities of Azure OpenAI Service for post-call transcription and analysis. You\'ve successfully created the necessary resources, configured models, and set up environment variables. Furthermore, you\'ve executed sentiment analysis and summarization tasks on call center transcriptions, gaining firsthand experience in utilizing these tools effectively.

By the end of this lab, you will have gained practical experience in harnessing Azure OpenAI Service for efficient post-call transcription and analysis, which can be invaluable for enhancing customer service and decision-making processes.

**Important Note: Please do not delete the Resource group and Azure OpenAI Service (Azure-openai-testXX). The same Resource group and AOAI service will be used throughout all the labs.**