Microsoft Azure

Demo Script: Azure AI Studio Deep Dive

Intro | Demo <"MSFT and GOOG Environmental sustainability initiatives">

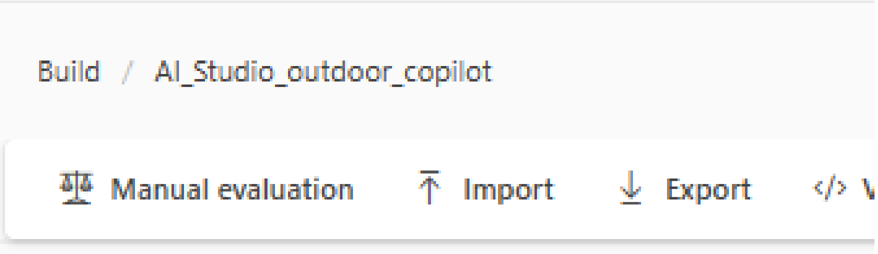
We have all seen powerful co-pilots like BingChat, M365, and GitHub Copilot. These are all built and Azure and are some of the world's most complex workloads. Through this session today, we are going to build our own custom co-pilot using Azure AI Studio.
Goal is to help BOKF familiarize with building custom Co-pilots using Azure AI Studio and make their current custom applications AI infused.

<<AI Resource>> The first thing that I will do as a part of building my application, is creating my project. Through this, I created an Azure AI resource. The Azure AI resource helps me connect to, create, and later as you will see, manage all the AI related assets that go into my project. These include resources like Azure Machine Learning, Azure OpenAI Service, Azure AI Speech, and Azure AI Search. This minimizes setup challenges, enabling users to start building quickly and seamlessly.

Task 1: Create a new project inside the AI Hub

- Open a new browser window
- Login to https://ai.azure.com use your BokF credentials
- Under AI Hubs select the AI Hub (there should be only one Ai Hub) and create a new project any name
- Once the AI project is created, verify if you are able to see the

Summary:
I have created a project, In the setting tabs, you can see all the different connections within this project, the project members that have access, how resources are configured including AOAI. Withing the deployments tab, we have all the different models that are deployed within my AOAI resource. They are ready for me to use. If I need additional deployments, I can create those here and use within my project.



Task 2: Start using the Chat playground

- Click on the “chat” button under playground
- Ask any random question

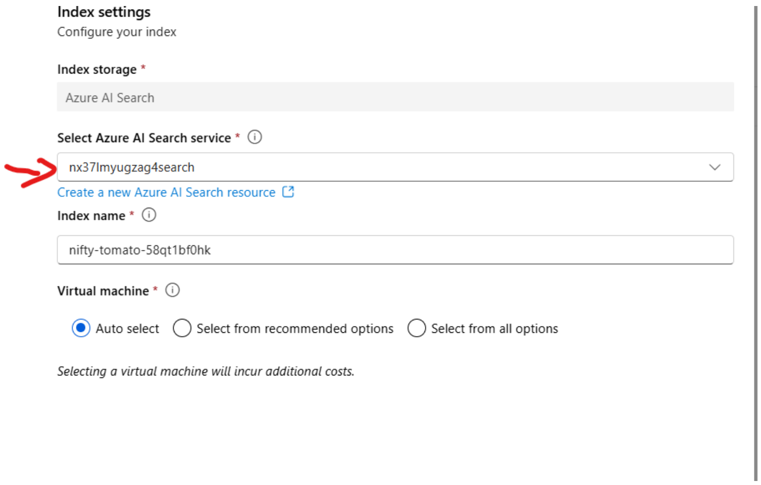
<<Playground>> I am going to go to the project playground now, which is a great place to get started. I am using a GPT-4o (might be different for some of you)model. Initially, when I asked specific questions about my data, the model didn't know because it wasn't grounded in my data. As expected, it’s saying that I do not have access to the pricing information. So, let’s go ahead and fix that now.

Task 3: Add your data to provide context to the LLM model

- Click on the “data” button under components
- Click on new data and in the Data source dropdown select “upload files and folder” option
- Choose the folder option and browse to the folder on your lapton to where the files where downloaded and choose the folder option
- Wait till the upload is completed
- Give a name to the Data Source

Task 4: Create the Vector search Index

- Click on indexes under components
- Click on new index and in the Data source select “Data from AI studio”
- Select the data source that was created in the previous task and click next
- In the Index settings screen, select the Azure AI search service from the dropdown (there should be only one search service). Leave the rest of the settings as default and click next



Index settings
Configure your index

Index storage *
Azure AI Search

Select Azure AI Search service *
nx37lmyugzag4search
Create a new Azure AI Search resource

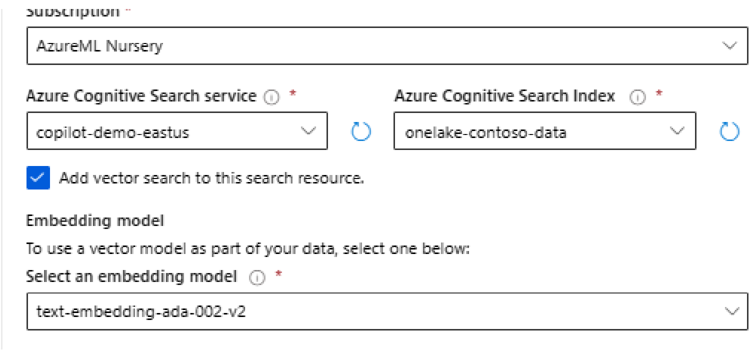
Index name *
nifty-tomato-58qt1bf0hk

Virtual machine *
● Auto select ○ Select from recommended options ○ Select from all options

Selecting a virtual machine will incur additional costs.

- In the search settings screen select the “add vector search to this search resource” and select the embedding model service name (there should be only one in the drop down)
- Review and Click on create

<<Add data>> I am going to ground this model with my data, using Azure AI Search. By ingesting our data from OneLake and using a Vector Index through Azure AI Search, we've solved this problem. As you can see, this time when I ask the question, it snows the answer. It also references which product document it is retrieving that information from. Not only have we grounded this model, but we have also enabled multilingual responses.



Subscription *

AzureML Nursery

Azure Cognitive Search service * copilot-demo-eastus

Azure Cognitive Search Index * onelake-contoso-data

☑ Add vector search to this search resource.

Embedding model

To use a vector model as part of your data, select one below:

Select an embedding model * text-embedding-ada-002-v2

<<OneLake/Fabric>> OneLake is supported in the runtime and therefore it is fully compatible across Azure AI Studio. We can use it in the playground, as a source for FT and within my orchestration flow as you will see later.

Now that I understand how this model works, let's take a look at some of the other tools that we have within the playground.

<<Prompt Catalog>> As we can see here, my system message or prompt is basic. The quality and relevance of the response generated by the LLM is heavily dependent on the quality of the prompt. So, prompts play a critical role in customizing LLMs to ensure that the responses of the model meet the requirements of a custom use case. I can use the Prompt Catalog to select from prompts across different industries.

<<Manual Test>> In the Playground, I have been having a 1:1 conversation with this model. For me to understand more about the behavior of my application grounded with my data, I want to be able to ask several questions and compare the results. For that we can use our manual evaluation tool. Both the system message and my vector index have been carried over. Let's change this to the GPT-4o (might be different for some of you) model. I can input questions directly in the tool. Optionally, I can add expected responses to compare with the output from the LLM. What could be quicker is if you import an entire dataset. As you see, I have already gone ahead and done that. I had a dataset that had around 10 questions, along with expected response that

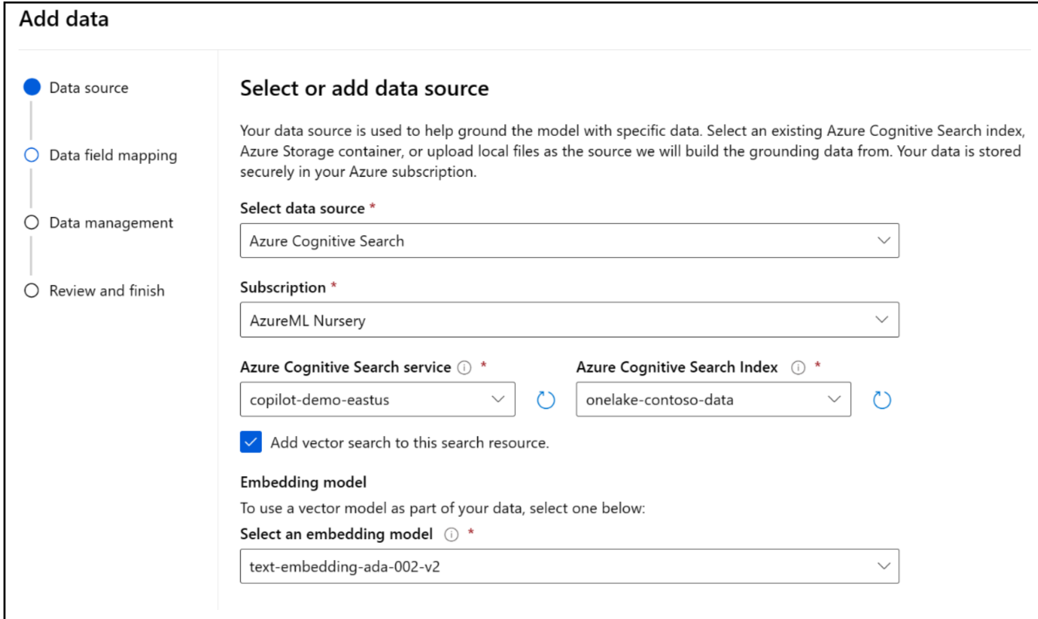
I know to be accurate. I am going to run all of them as a manual evaluation. After the test runs (which is pretty fast), I can quickly review all the outputs in conjunction with my dataset and its expected results. Now, let's analyze these responses. Many of them appear highly accurate. For those that meet the mark, I can express my preference. But, if a response falls short, I have the option to flag it for further attention. The effort invested in this process doesn't go to waste. I can easily export these results for sharing purposes. Additionally, I can preserve this dataset, utilizing it later in my project for a more comprehensive evaluation as we will see later.

Act Two: Tearing into Data & Search Structures | *Demo <ADD YOUR DATA, SEMANTIC RANKER & VECTOR SEARCH>*

Now that you have an idea of how the playground works, I want to go back and talk a bit more about Azure AI Search that we used to ground our model.

Under-the-covers, the playground runs a sophisticated ingestion pipeline that cracks open the documents, runs a chunking strategy, embeds the data using an LLM and inserts the embeddings into an Azure AI Search. **First, we have vector search.** Vector search is great at understanding relationships between words. For example, it can understand the relationship between a tent and camping. However, it doesn’t do a good job with exact words. For example – phone numbers, email addresses, or product names. For that good old keyword search works much better. **With hybrid search**, you can use both keyword search (when users need exact info) and vector search (which better understands user intent). **Last, we have semantic ranker.** This is the search that rules them all. It builds on top of hybrid by adding a ranker that makes sure the best information gets surfaced to the top of our search. Now let's see this in action.

Microsoft Azure



Add data

- ● Data source
- ○ Data field mapping
- ○ Data management
- ○ Review and finish

Select or add data source

Your data source is used to help ground the model with specific data. Select an existing Azure Cognitive Search index, Azure Storage container, or upload local files as the source we will build the grounding data from. Your data is stored securely in your Azure subscription.

Select data source *: Azure Cognitive Search

Subscription *: AzureML Nursery

Azure Cognitive Search service *: copilot-demo-eastus

Azure Cognitive Search Index *: onelake-contoso-data

☑ Add vector search to this search resource.

Embedding model

To use a vector model as part of your data, select one below:

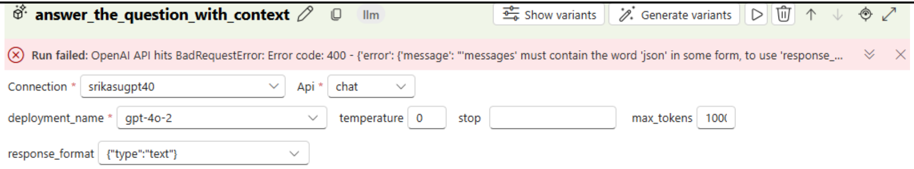
Select an embedding model *: text-embedding-ada-002-v2

Act Three: Testing, customizing, and deploying Proof of Concepts (PoCs)

Demo <PROMPT FLOW, EVALUATION, CONTINUOUS MONITORING >

Task 5: Repeat the same “chat with your data” scenario with Prompt flow

- Under tools select Prompt flow and click on create button
- Select the “multi round Q & A on your Data” from the gallery and click on clone
- Name the new flow appropriately and click on clone (it will take a few mins to deploy)
- After the clone is successful go ahead and start the compute session. This may take a few mins
- After the compute has successfully started, modify the promptflow steps
  - MLIndex setup
  - Answer the question with context



answer_the_question_with_context

Run failed: OpenAI API hits BadRequestError: Error code: 400 - {'error': {'message': "'messages' must contain the word 'json' in some form, to use 'response_...

Connection * srikasugpt40
Api * chat
deployment_name * gpt-4o-2
temperature 0
stop
max_tokens 100(
response_format {"type":"text"}

Task 5 (contd):

- Click on save and then click on “chat”

Now that you have seen so many exciting possibilities, the reality is a lot more complex. Customers want a way to add many data sources, to use other models alongside Azure OpenAI Service, and to test variations of meta prompts and models together.

Azure AI Studio offers users direct access to the orchestration engine that fuels this UI with prompt flow. The data sources and APIs created by the wizard are represented as nodes within a flow.

If I prefer to develop in a code-based environment, I can easily transition since the nodes of the flow are backed by files stored in my project storage. The integration with VS Code let’s me run the same flow in a pre-configured dev environment using the AI CLI.

In VS Code, I see folders for my code and data, and a shared folder that has the prompt flow I was working on in the studio. I can continue my flow development, and since I am editing the same files, the changes I make in VS Code will be reflected in the UI as well.

Developers face challenges with LLMs due to their non-deterministic nature. Minor changes in meta prompts can yield different results. To tackle this, developers require evaluation tools

I will be testing two meta prompts: one simple and another with instructions to promote safer behavior.

I am going to use the same dataset that we created earlier. It includes a set of questions and answers that I want to evaluate my flow against. Azure AI Studio gives me a comprehensive list of metrics that help me evaluate the impact of changes in my application. The platform provides a diverse set of pre-built, traditional machine learning and AI-assisted metrics for measuring answer quality and safety, including relevance, coherence, fluency, similarity, and groundedness (how grounded is the answer in the context provided).

With a few clicks, I have generated metrics to evaluate each variant of the meta prompt. Variant 0 was the basic prompt and Variant 1 was the safer prompt. In general, it looks like the safer prompt has slightly better metrics. If I want to dig deeper into the metrics for this variant, we can do that too in this instance view. Here, I can see a set of metrics, along with the information I need to pinpoint where the gaps are in my application for the safer variant. If I need more control over how these metrics are calculated - the evaluation process, powered by GPT-4o, operates as a prompt flow. This not only gives you transparency into how these metrics are calculated, but also complete control over defining your own metrics.

Let's look a bit more into deployment. With a few clicks, I can deploy to an endpoint, integrate it in my application. I can test my deployment, consume it in my application and I can enable data collection from the deployed endpoint. This enables me to monitor my application’s performance and safety using the same metrics as building my application.

Microsoft Azure

Content filters screen out harmful content, ensuring control and peace of mind in my application

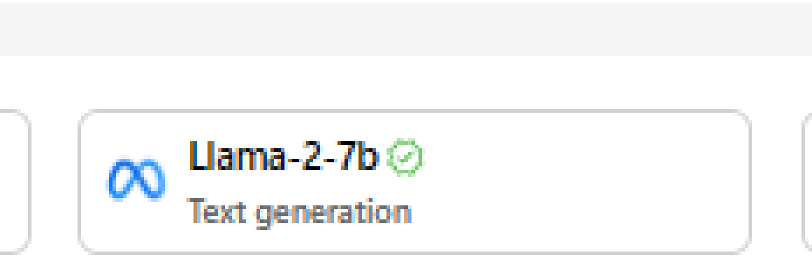
Act Four: Lifting with models | Demo <MODEL CATALOG, BENCHMARKING, MaaS, FINE-TUNING><MULTI-MODAL><DALLE3>

We have been using the GPT-4o (might be different for some of you) along with an ADA model for the embeddings used to ground our data. Let's say you also want to use other models in your application to reduce cost, or improve performance or accuracy in your application.

Imagine the flexibility to choose from thousands of open-source models With the benchmarking tool in AI Studio, you can benchmark and select the best model for your unique needs - like Meta's 70 billion parameter Llama-2 model.

Now that you know how to choose a model, let's talk about customizing that model for your scenario.

With Azure AI Studio, we are also introducing Model as a Service. You can easily fine-tune and deploy Llama and AOAI models with just your training data and an API call, while we manage the quota for you.



Microsoft Azure

Tailor-made AI solutions have never been this accessible.

Having seen a complete text-in, text-out scenario, let's look into what the future of generative AI unfolds …… which is multi-modality. Multi-modality enables me to craft dynamic applications incorporating images, text, speech, and even videos.

We are going to look at several scenarios:

In the first one, I can generate content for my Bank Marketing campaign <Completions >

I can then build on top of that, by also generating images in various Marketing conditions for my banking campaigns <Image> **(DALLE 3)**

This is where it gets exciting. Now let's go hiking. Thanks to the cutting-edge GPT-4 Turbo with Vision, we've taken personalization to new heights. Seamlessly integrating video content, this technology empowers my application to analyze and curate personalized product suggestions and create outdoor itineraries. Witness the magic as the enhanced GPT-4 Turbo model precisely pinpoints the ideal hiking trail at Yellowstone National Park and crafts a itinerary along with a packing list for your journey. And that's not all – with the power of Azure AI Speech, I can enable speech-to-text functionality in the playground, enhancing your experience further. <Mulitmodal> **(GPT-4 Turbo with Vision)**

Microsoft Azure

There is more to see here, with the integration of Custom Neural Voice into Azure AI Studio, you can create a custom voice solution that can adapt across languages and speaking styles. **(CNV)**

Our Azure AI Studio tour has ended, but your journey is just starting! We're thrilled to equip you with the tools to safely build Generative AI applications using Azure AI Studio.