Lab Guide:
Let's Prompt Tune with watsonx.ai

The Art of IBM's AI Customization: *Your Data, Your Tuned Model*

Authors: Joost Vos, joost.vos@ibm.com

Marc Decker, marc.decker@nl.ibm.com

1. "*Prompt Tuning*", an introduction into Large Language Model optimization:

Author: Kim Martineau[1]

Prompt-tuning is an *efficient, low-cost way of adapting an AI foundation model* to new downstream tasks without retraining the model and updating its weights. Foundation models are set to usher in the next wave of AI enterprise applications. These large, reusable models have been pretrained on the vast knowledge of the internet, making them easier to customize for things like analyzing legal contracts or detecting fraud in financial documents.

Until recently, fine-tuning was the best way to redeploy one of these pretrained models for specialized tasks. You gathered and labeled examples of the target task and fine-tuned your model rather than train an entirely new one from scratch. But a simpler, more energy-efficient technique has emerged as foundation models grow relentlessly larger: prompt-tuning.

In prompt-tuning, the best prompts are fed to your AI model to give it task-specific context. The prompts can be extra words introduced by a human, or AI-generated numbers introduced into the model's embedding layer. Like crossword puzzle clues, both prompt types guide the model toward a desired decision or prediction. Prompt-tuning allows a company with limited data to tailor a massive model to a narrow task. It also eliminates the need to update the model's billions of weights, or parameters.

Redeploying an AI model without retraining it can cut computing and energy use by at least 1,000 times, saving thousands of dollars, said IBM's David Cox, head of Exploratory AI Research and co-director of the MIT-IBM Watson AI Lab. "With prompt-tuning, you can rapidly spin up a powerful model for your particular needs," he said. "It also lets you move faster and experiment."

Prompt-tuning originated with large language models but has since expanded to other foundation models, like transformers that handle other sequential data types, including audio and video. Prompts may be snippets of text, streams of speech, or blocks of pixels in a still image or video.

"It's a fast and sustainable way of extracting knowledge from these large models," said IBM's Ramewsar Panda, an expert on prompt-tuning at the MIT-IBM lab. *"We don't touch the model. It's frozen."*

---

[1] https://research.ibm.com/blog/what-is-ai-prompt-tuning

1.1 watsonx.ai

Watsonx.ai is a core component of watsonx, IBM's enterprise-ready AI and data platform that's designed to multiply the impact of AI across an enterprise.

The watsonx platform has three powerful components:

- watsonx.ai studio for new foundation models, generative AI and Machine Learning (traditional AI)
- watsonx.data fit-for-purpose data store that provides the flexibility of a data lake with the performance of a data warehouse
- watsonx.governance toolkit, which enables AI workflows that are built with responsibility, transparency, and explainability.

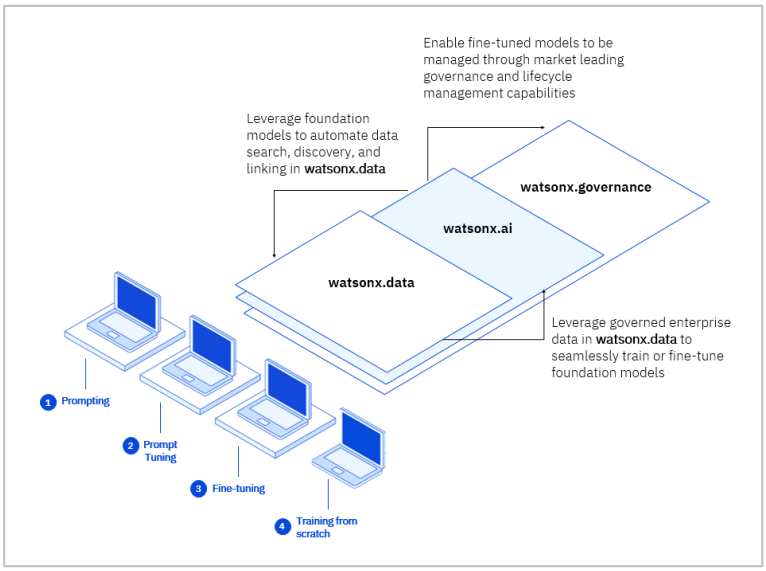


Figure 1: IBM watsonx platform

The watsonx.ai component (the focus of this lab) makes it possible for enterprises to train, validate, tune, and deploy AI models – both traditional AI and generative AI. With watsonx.ai, enterprises can leverage their existing traditional AI investments as well as exploit the innovations and the potential of generative AI using foundation models to bring advanced automation and AI-infused applications to reduce cost, improve efficiency, scale, and accelerate the impact of AI across their organizations.

1.2 Prerequisite to this lab

This IBM watsonx.ai Technical Hands-on day 2 Lab is a **follow up** on the IBM watsonx.ai – Let's create a conversational search AI agent. It is assumed that you have completed this first Hands-on lab. Detailed instructions and screen flow available there are not repeated here.

1.3 What we will learn in this lab

In this lab, you will be learning how to optimize the performance of a pre-trained large language model available from the watsonx.ai model catalog by means of **prompt tuning.** In this lab you will use the following IBM Cloud components:

1. **watsonx.ai** – IBM's AI studio to train, validate, tune and deploy AI models.
2. **IBM Watson Studio** – IBM Watson® Studio empowers data scientists, developers and analysts to build, run and manage AI models, and optimize decisions anywhere.
3. **Watson Machine Learning** – IBM's service to deploy, manage and integrate generative AI prompt tuned models as well as traditional machine learning models into your applications and services in as little as one click.

Although each service can be deployed on-premises, in this lab we will make use of cloud services only.

The documentation for watsonx.ai can be found at:

https://www.ibm.com/docs/en/watsonx-as-a-service

1.4 Pre-trained Large Language Models

A large language model is trained in massive amounts of text data. The source data oftentimes consist of internet data. Pre-trained models are trained on very diverse datasets. Additionally, models can als be optimized for a specific type of application. For the latter, you will often observe “chat” and “instruct”-suffixes to the names of large language models. These labels indicate the particular task for which they are fine-tuned. **Chat** means the model is optimized for conversational tasks, like application in a chatbot. **Instruct** indicates that the model is optimized for carrying out particular instruction and the follows upon that. There is lots of variations in datasets, language coverage, as well as (specialized) tasks on which language models have been trained. Some are specifically trained on computer code data, some are trained on plain internet data, some are trained on specific domains like financial news. That’s why so many pre-trained large language models already exist. As of the writing of this lab guide, through Huggingface the number of publicly available open source models that are optimized for the task of text generation exceeds the number ***80,000 different models*** *(figure 1)*.

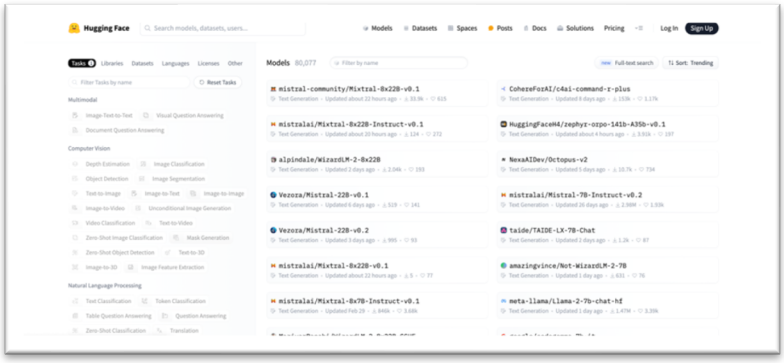


Figure 1 - Number of open source text generation models on HuggingFace (Apr 2024)

This tells us a couple of things:

- Large language models are found ***extremely versatile*** and can be applied in a wide range of software applications
- Popularity of open source movement
- There is **diversity** in models, because they:
    - are trained on ***different source datasets***
    - vary in ***language support***
    - differ in **architecture**
    - are ***optimized for diverse specific tasks***
    - differ in ***size***
    - have the potential to run ***on-prem or on edge***

1.5 There is *'no model that will rule them all'*

There isn't one single model or approach that can solve every business problem. Instead, people use different models and techniques depending on the specific task at hand. If we talk about large language model optimization, there are four ways to optimization. The list below provides the ways in ascending complexity:

- **Prompt engineering:** the process of writing, refining, and optimizing inputs to encourage generative AI systems to create specific, high-quality outputs (also covered in the *first workshop*)
- Prompt-tuning: Adjusts the content of the prompt that is passed to the model to guide the model to generate output that matches a pattern you specify. The underlying foundation model and its parameters are not edited. Only the prompt input is altered. It represents a very cost effective and quick method to adapt a pre-trained large language model to your specific business needs. Perhaps more importantly, prompt tuning may allow a *smaller model to perform even better than a bigger model* for specific prompt tuned tasks. This allows creating higher accuracy at lower inference cost at production.
- Fine-tuning: Changes the parameters of the underlying foundation model to guide the model to generate output that is optimized for a task.
- Training: building a model from scratch.

Prompt engineering, prompt tuning and fine-tuning are all language model optimization capabilities supported by IBMs watsonx.ai.

2. Prompt-tuning

Prompt-tuning is a technique used to adapt large language models (LLMs) to perform specific tasks without the need to retrain the entire model. In the possibilities of prompting, it follows the manual zero-shot prompting and few-shot prompting (figure 2). For Prompt-tuning, you typically need 100s-1000s labeled data points as example data. Next to that, you need to have a picture on what you want the model to give as an answer. Both the input as well as the output are wrapped in the data set with labeled data points. Also here, the widely known adage is true: *"garbage in = garbage out"*. The higher the quality of your labeled data, the more accurate the performance of your tune will be. Please find the *typical steps of prompt tuning* below figure 2.

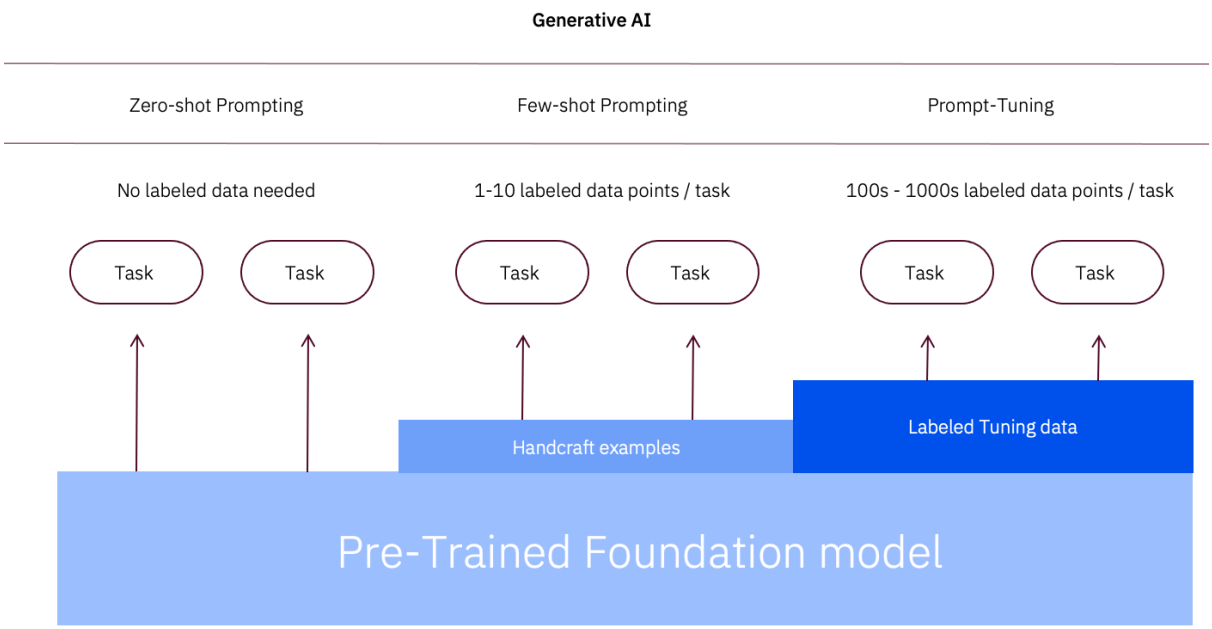


Figure 2 - prompting in ascending complexity.

2.1 How it works.

Foundation models are sensitive to the input that you give them. How you prompt the model can introduce context that the model will use to tailor its generated output. Prompt engineering to find the right prompt often works well. However, it can be time-consuming, error-prone, and its effectiveness can be restricted by the context window length that is allowed by the underlying model.

Prompt-tuning a model in the Tuning Studio applies machine learning to the task of prompt engineering, hence the use of ***Watson Machine Learning***. Instead of adding words to the input itself, prompt-tuning is a method for finding a sequence of values that, when added as a prefix to the input text, improve the model's ability to generate the output you want. This sequence of values is called a ***prompt vector***.

Normally, words in the prompt are vectorized by the model. ***Prompt-tuning bypasses the model's text-vectorization process and instead crafts a prompt vector directly. This changeable prompt vector is concatenated to the vectorized input text and the two are passed as one input to the embedding layer of the model.*** Values from this crafted prompt vector affect the word embedding weights that are set by the model and influence the words that the model chooses to add to the output.

To find the best values for the prompt vector, you run a tuning experiment. You demonstrate the type of output that you want for a corresponding input by ***providing the model with input and output example pairs in training data***. With each training run of the experiment, the generated output is compared to the training data output. Based on what it learns from differences between the two, the experiment adjusts the values in the prompt vector. After many runs through the training data, the model finds the prompt vector that works best.

You can choose to start the training process by providing text that is vectorized by the experiment. Or you can let the experiment use random values in the prompt vector. Either way, unless the initial values are exactly right, they will be changed repeatedly as part of the training process. Providing your own initialization text can help the experiment reach a good result more quickly.

The result of the experiment is a tuned version of the underlying model. You submit input to the tuned model for inferencing and the model generates output that follows the tuned-for pattern.

Source: https://www.ibm.com/docs/en/watsonx-as-a-service?topic=studio-methods-tuning

2.2 A stepwise overview of Prompt Tuning

Video: https://video.ibm.com/channel/23952663/video/wx-tuning-studio

Step 1 – Model selection and understanding its capabilities.

Before you start prompt tuning, it is essential to understand the capabilities and limitations of the LLM you are working with. Familiarize yourself with the model's architecture, pre-training data, and the types of tasks it is designed to perform. ***Be sure to inspect the model card in watsonx.ai!***

Step 2 – Test the model and designing the prompt.

After model selection, it's a good practice to test the selected base model. Do some manual test driving in the prompt lab to get a feeling what the model its out-of-the-box performance for your task is. Evaluate both the zero-shot as well as few-shot approaches. In a client setting, you might do some model evaluations with a client's domain expert – since they know their content/industry best. Try to design a prompt and/or instruction (that is: the specific task for the model to carry out) that is clear, concise, and task aligned. The prompt should provide context to the model and steer it toward generating the desired output. Consider the following when designing your prompt:

- **Clarity:** The prompt should be straightforward and unambiguous.
- **Conciseness:** Avoid unnecessary information that could confuse the model.
- **Alignment:** Ensure the prompt aligns with the desired output in terms of style, tone, and content.

Step 3 – Collect example data containing input-output pairs.

Prepare a set of prompt examples to use to tune the model. The examples must contain the type of input that the model will need to process at run time and the appropriate output for the model to generate in response. Prompt input-and-output example pairs are sometimes also referred to as samples or records. Use at minimum 50 examples. Watsonx.ai supports tuning with training data containing up to 10,000 examples.

Step 4 – Setting the task for prompt-tune.

IN watsonx.ai, you can select the task for which you would like to prompt-tune your model. You can select from the options of:

- Classification *predicts categorical labels from features.*
- Summarization *generates text that describes the main ideas that are expressed in a body of text.*
- Generation *generates text such as a promotional email.*

Step 5 – Setting the tuning parameters.

You can set the parameters for your prompt-tune your model or leave them unchanged while working on the labs. Your instructor will provide some context behind the different parameters. You can find detailed documentation here:

https://www.ibm.com/docs/en/watsonx-as-a-service?topic=model-tuning-parameters#initialize

Step 6 – Deploy the tuned model to a deployment space.

When the prompt-tune is ready, you can promote the model to a deployment space. This is needed to be able to test and validate the prompt tuned model in the watsonx.ai Studio. Once promoted to the deployment stage, you can also inference the model from your application.

Step 7 – Test the tuned model.

It is a good habit to check the prompt-tuned model before consuming it into a production environment. Feel free the experiment with different inputs on which the tune was trained to test whether the LLM outputs the results you aimed for.

Step 8 – Integrate into your application of choice.

Once satisfied, you can integrate your model from the deployment space into production applications.

3. The Labs – Classification, Summarization, Generation

3.1 Disclaimer

IBM watsonx.ai is developed and released in an agile manner. In addition to constantly adding new capabilities, the web interface is likely to change over time. Therefore, the screenshots used in this lab may not always look exactly like what you see. You can expect to encounter some of the following:

- Additional foundation models in the library list
- Removal of deprecated foundation models in the library list
- Additional foundation models available for prompt tuning
- Changes in the user interface (location of buttons, text for various fields)
- Additional tabs/buttons

3.2 About the lab

In this lab, you will be introduced to the following watsonx.ai capabilities:

- Explore the Tuning Studio on the IBM watsonx.ai console.
- Perform prompt tuning on a foundation model using a set of labeled data.

3.2 Prerequisites & Getting Started

You will need an IBM Cloud account to gain access to the TechZone account that hosts the various Watson and watsonx services used in this lab. To complete this lab, you will need access to watsonx.ai. Your instructor will provide the login credentials. Next to that, you'll need the following:

A. A **text editor (notepad++, BBEdit)** – can't live without it ;-), you'll be generating some JSONs today.
B. All document data can be found in the following git-repository:

   https://github.com/joost-vos/watsonx-ai-prompt-tuning

C. Data directory:

   https://github.com/joost-vos/watsonx-ai-prompt-tuning/tree/main/prompt-tuning-data

D. Access to one or more generative AI webservices

3.3 The Prompt-Tunes!

You are going to prepare two (or three if you have time!) prompt-tunes. Each tune represents one of the tasks for which you can perform a tune. These tasks are:

- Classification
- Summarization
- Generation

You will be working your way through each of these tasks. For the first prompt tune – ***classification*** – , you will be getting everything you need to get from start to end - ***Easy peasy lemon squeezy***. In Dutch we'd say:"Appeltje, Eitje". The second one – ***summarization*** - will become a bit more difficult. We give you a partial 'howto' to set your creativity and problem-solving skills in motion. For the last task - generation – you will be working from zero to hero ***on your own***. Why? This reflects your real-life situation when interacting with your clients. Make use of the pointers you've obtained from the first two exercises and the guidance of your instructors. If you are stuck, ask your instructor for help! It is good to know that you find very good and detailed documentation here:

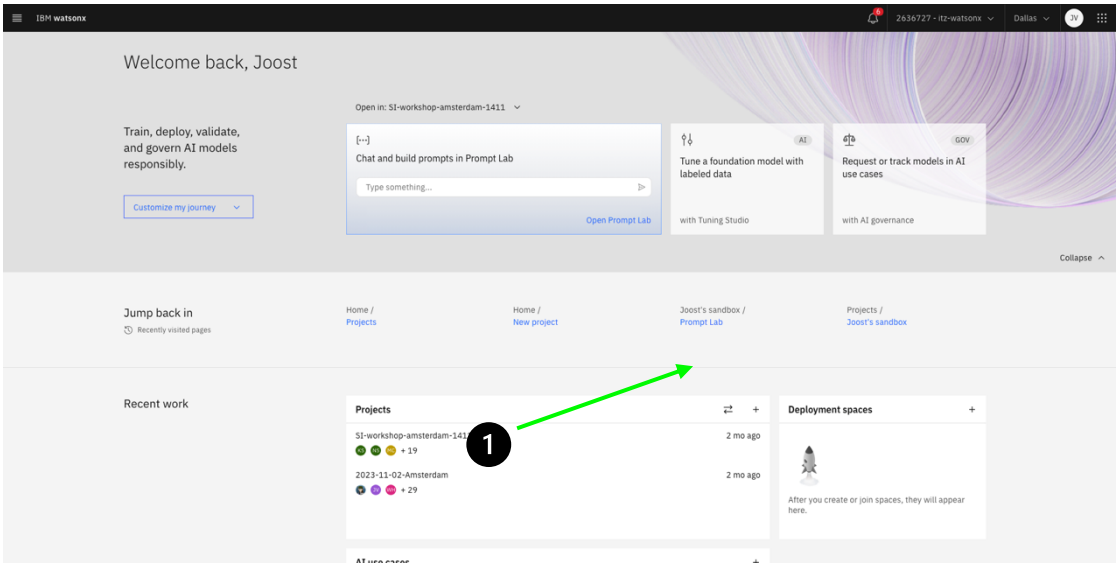
https://www.ibm.com/docs/en/watsonx-as-a-service

3.3 Exercise 1 - Classification

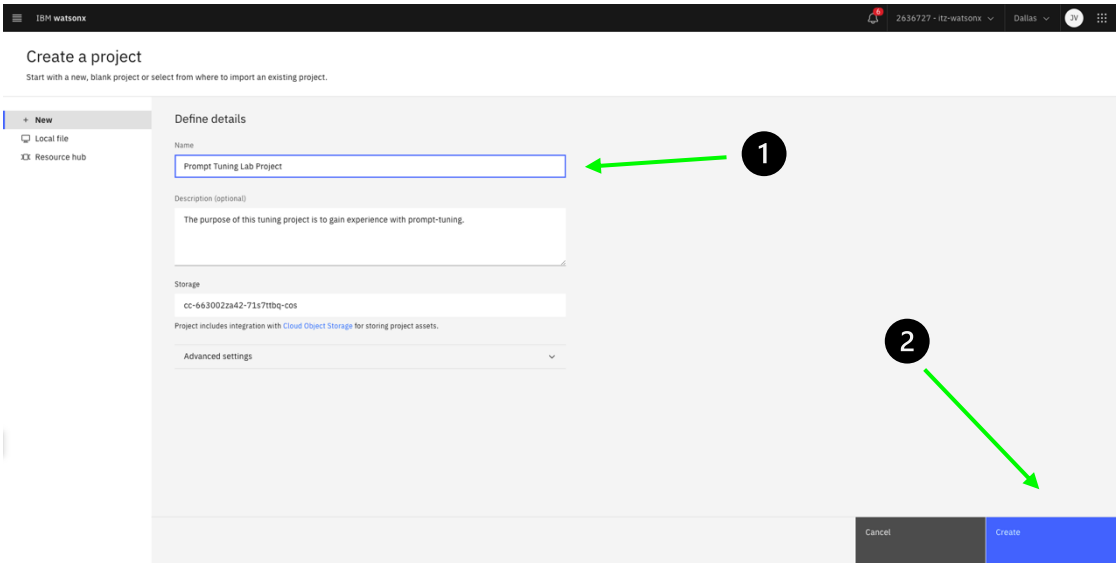
Classification Case: You are working on a fictious case for a fictious client called **LemonFresh**. LemonFresh is an online distributor for citrus fruits servicing one of the coldest and remote areas in Canada. As citrus fruits are fresh products with relatively short shelf-life, they aim to optimize citrus fruit supply and customer demand. They know that their consumers are very actively posting about their fruit consumption on social media, particularly on the FruitBowl platform. FruitBowl is a fruit consumer targeted social media platform that allows people to exchange their fruit eating experiences. LemonFresh aims to gather consumer insights into ***what fruits are eaten***. As a first step, the marketing director wants to know **how many posts that fruit lovers share on FruitBowl mention citrus fruits**. They officially obtained an anonymized dataset through the FruitBowl data services for which its users consented. The prompt tuning dataset you are using is devoid of any user data. Your client heard good stories about watsonx and the marketing director insists that you will be using IBMs watsonx platform. As a business partner, you've been assigned the task to classify fruit lover posts on whether they post about citrus fruits or not. ***Off we go!***

Exercise 1 – the classifier

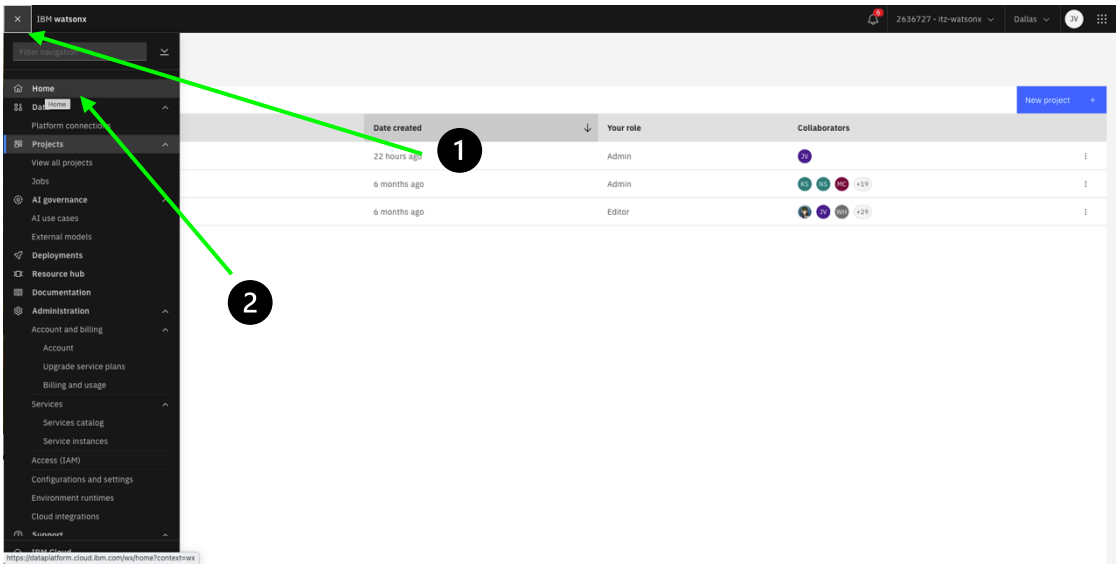
1. Create project, by clicking **+,** as indicated below



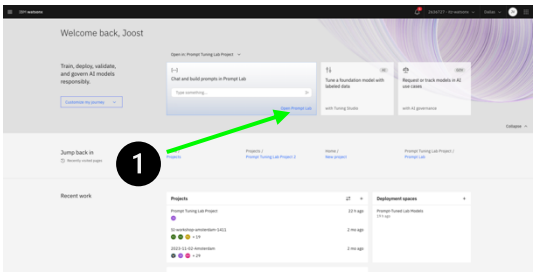
2. Give the project a **name** and an *optional* **description**, click on **Create.**

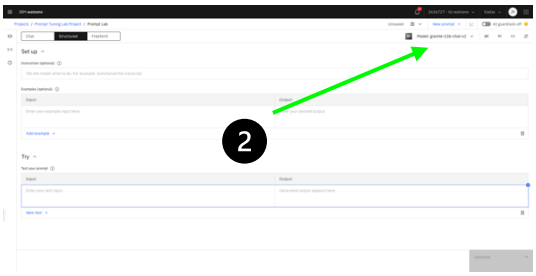


After project creation, you'll be guided to the **projects** page. From there navigate to the Prompt Lab by navigating to the **hamburger menu icon** in the top bar on the left hand side. Click on that, and next click on **Home**.

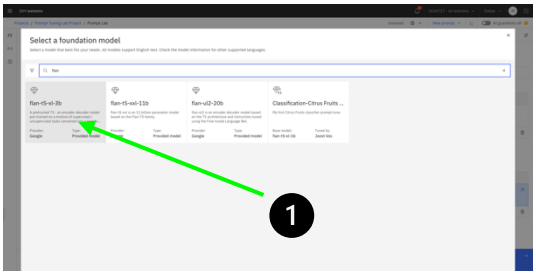


3. Open the **Prompt Lab** from the watsonx main page. In this lab we will be tuning the Flan T5 model.



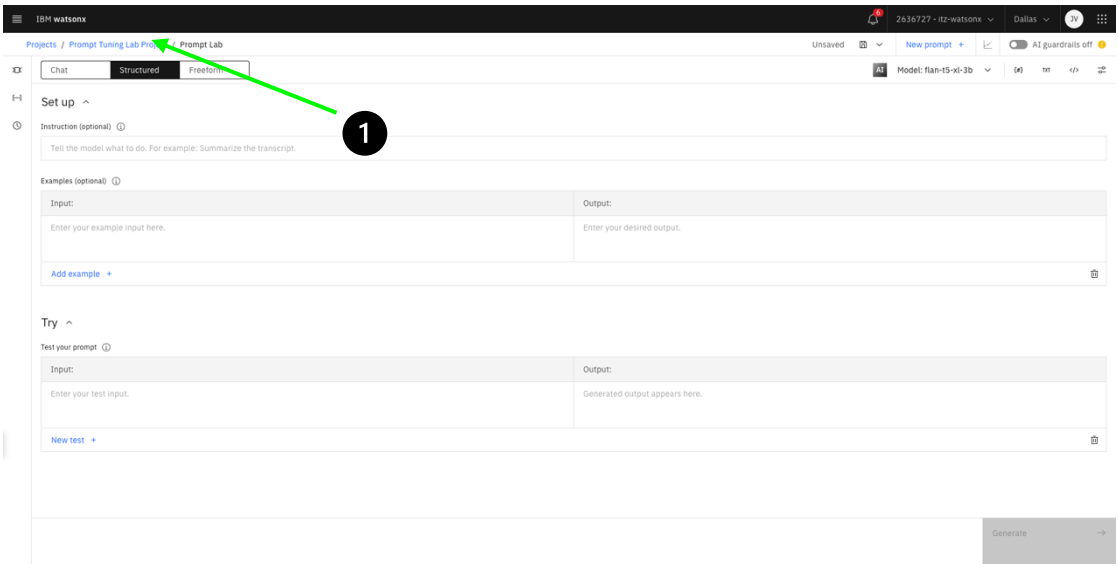


4. First select the **flan-t5-xl-3b** model. Be sure to inspect its **model card**. And click **Select Model** to continue.



After selecting the model of choice, you'll return to the prompt lab. You are ready for a test drive of the selected model.

5. Be sure you are in the Prompt Studio's Structured mode.



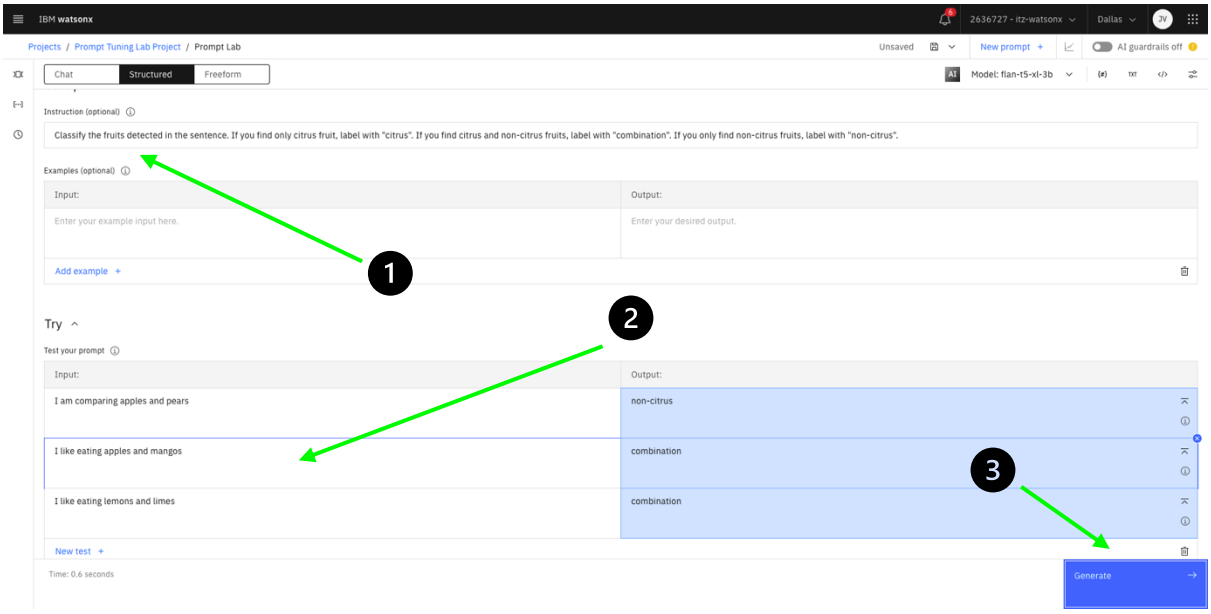
6. For evaluating the base model, use the following **instruction** and **input**:

Instruction: *Classify the fruits detected in the sentence. If you find only citrus fruit, label with "citrus". If you find citrus and non-citrus fruits, label with "combination". If you only find non-citrus fruits, label with "non-citrus".*

And the following 3 Inputs:

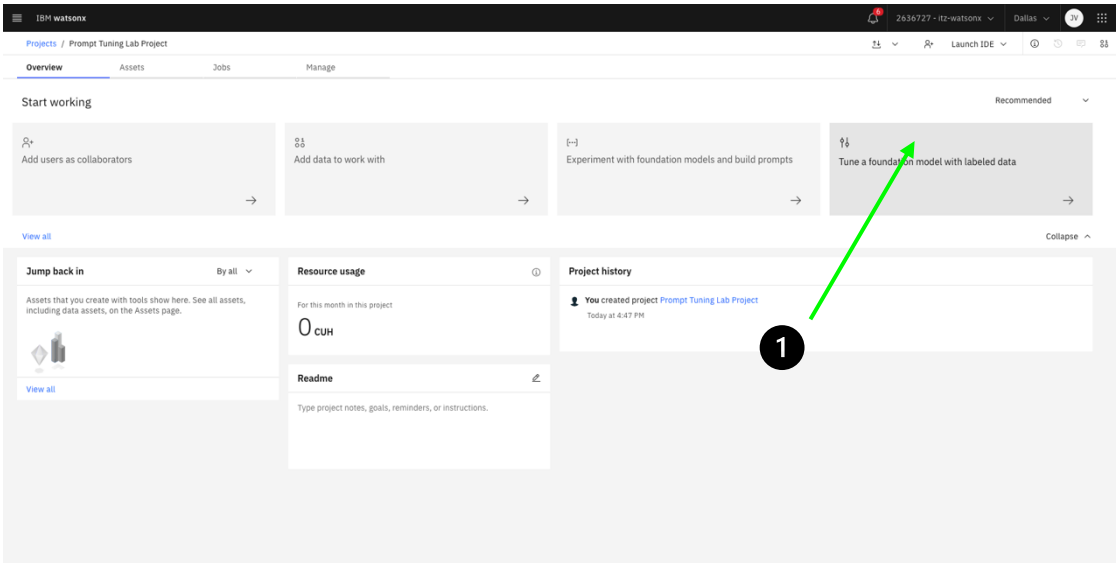
I am comparing apples and pears
I like eating apples and mangos
I like eating lemons and limes

Hit **Generate**

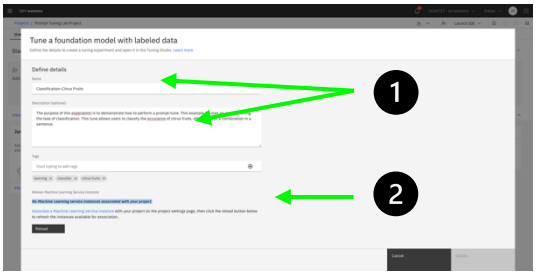


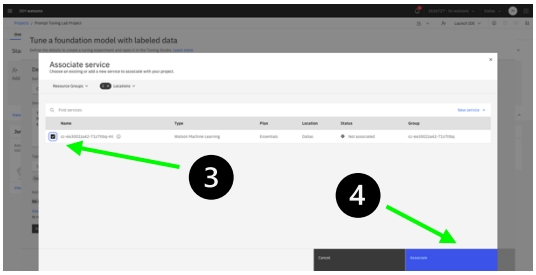
As you can see, the flan-t5-xl-3b base model only got the first input correct. There's room for improvement! ***Let's tune the model!***

7. Click on the name of the project in the **breadcrumb bar**. You will be guided to your **project page.** To start a prompt-tune click the tile "**Tune a foundation model with labeled data**".



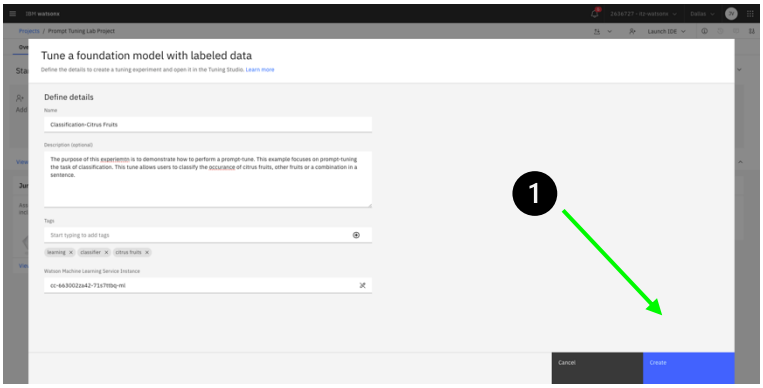
8. Give your prompt-tune a **name** and an *optional* **description** and *optional* **tags**. Next, associate a Watson machine Learning service by clicking on the link in blue. Then tick mark the Watson Machine Learning Service from the list and then click **Associate**.



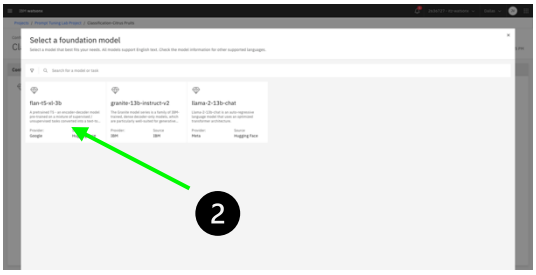


Note: *if you already associated a Watson Machine Learning instance previously, you don't need to repeat this.*

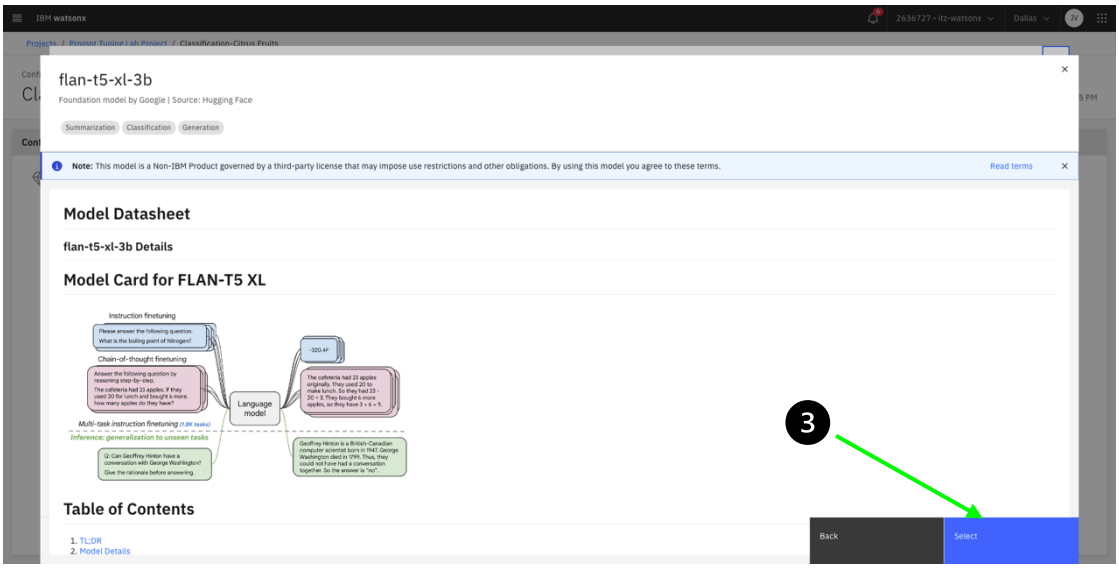
Now, you are ready to go. Click **Create**.



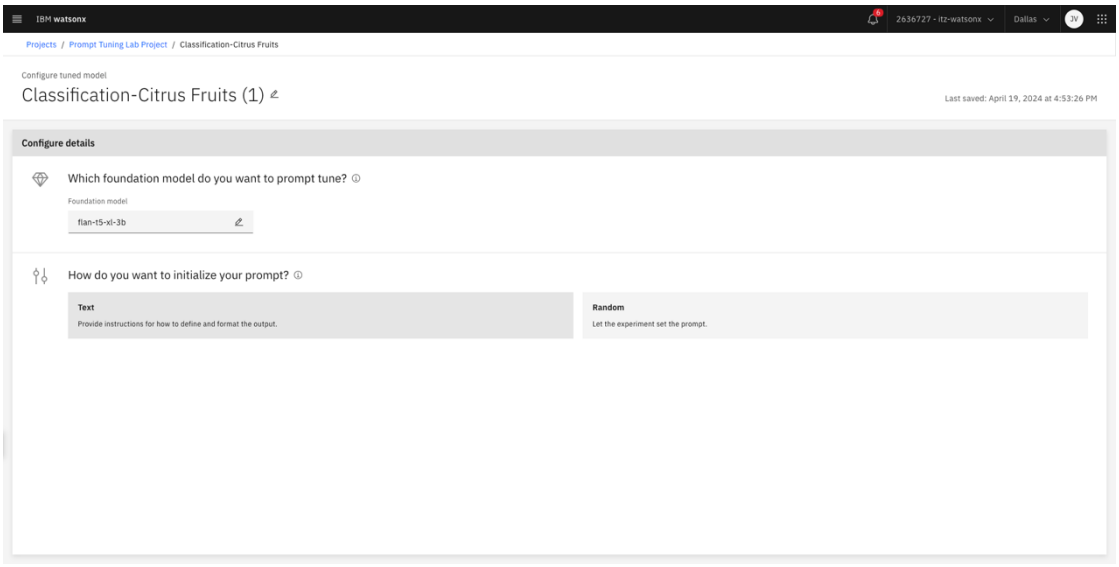
9. Select the foundation model you want to prompt tune. We will be using the **flan-t5-xl-3b** model.



Click on **Select**



10. Select **Text** from “How do you want to initialize the prompt?”



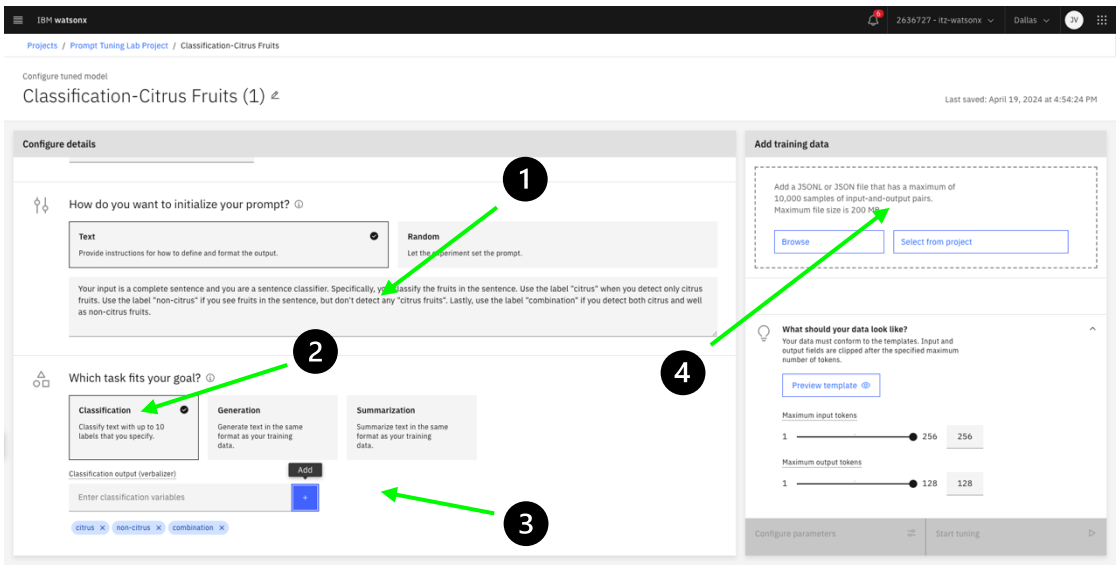
11. Either copy and paste the following instruction, or you can create one yourself.
    1. **Instruction:** *Your input is a complete sentence, and you are a sentence classifier. Specifically, you classify the fruits in the sentence. Use the label "citrus" when you detect only citrus fruits. Use the label "non-citrus" if you see fruits in the sentence, but don't detect any "citrus fruits". Lastly, use the label "combination" if you detect both citrus and well as non-citrus fruits.*

    2. Select the **Classification** task tile

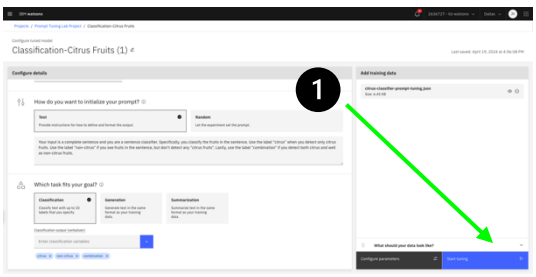
    3. Enter your classification labels: **citrus, non-citrus, combination.**

    4. **Download the JSON with training data here:**
    https://github.com/joost-vos/watsonx-ai-prompt-tuning/blob/main/prompt-tuning-data/citrus-classifier-prompt-tuning.json
    Drag and drop this file to the dotted area underneath the Add training data label

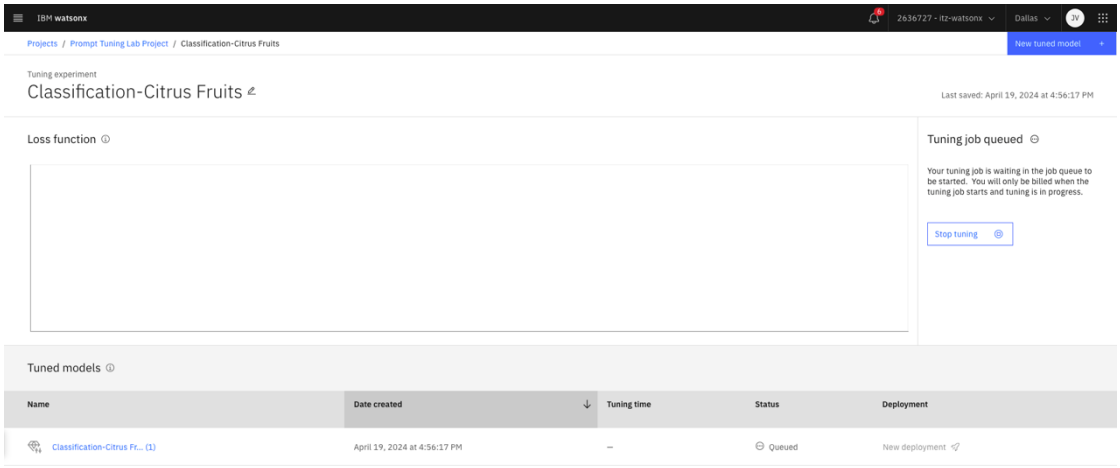
    5. **Optional: you can modify the maximum output tokens to 20. It'll force the model to just output the classification labels**



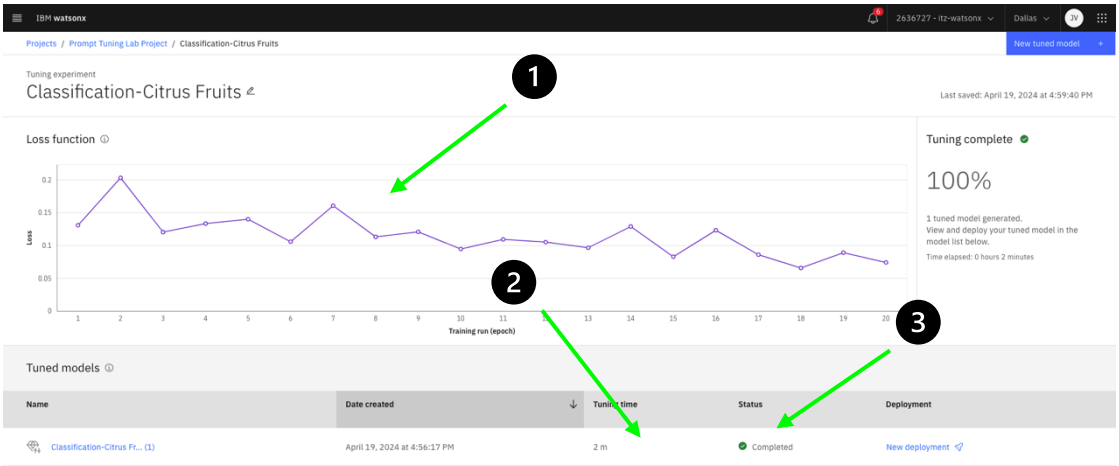
12. Next, click the **Start tuning** button



13. Your prompt-tune will sequentially be placed in the **queue** and subsequently processed. This will about 5-10 minutes of time to complete.

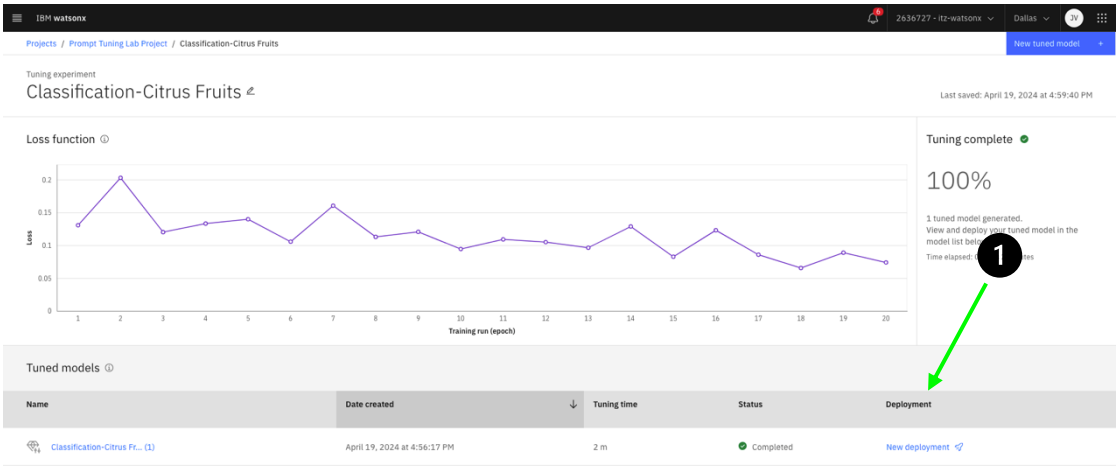


14. When done, please appreciate the **Loss Function**, **tuning time**, and **status**. The lower the final value for the loss function, the better your model will perform as assessed on the training data. If your tune worked out fine, the status should indicate **"completed"**.

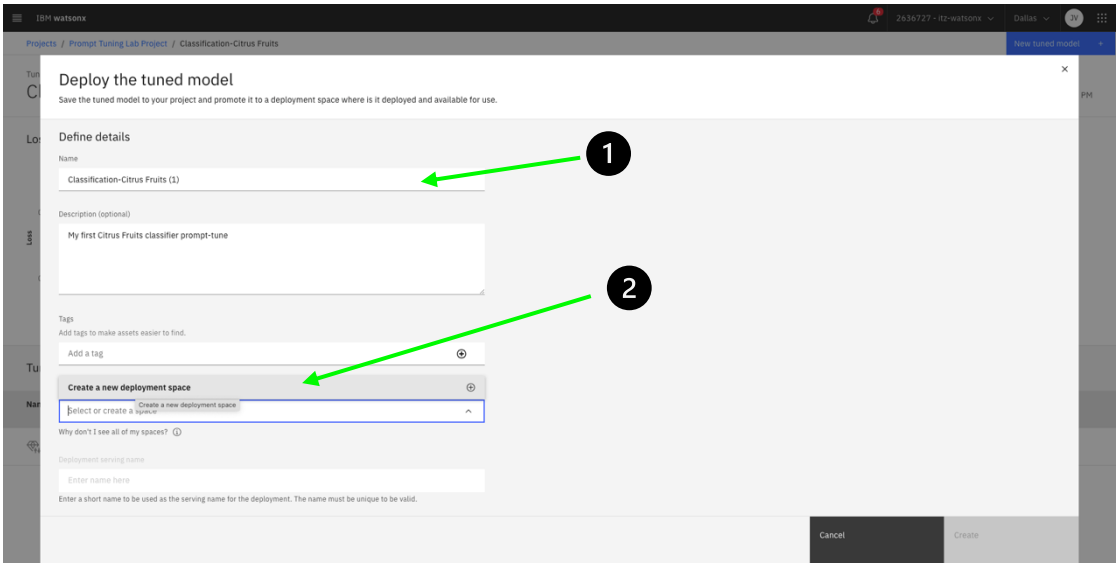


Now it is time to deploy the model. After deployment, you can (1) evaluate the prompt-tuned model in the Prompt Lab and (2) inference the model from externally, and thus ready to consume in your applications.

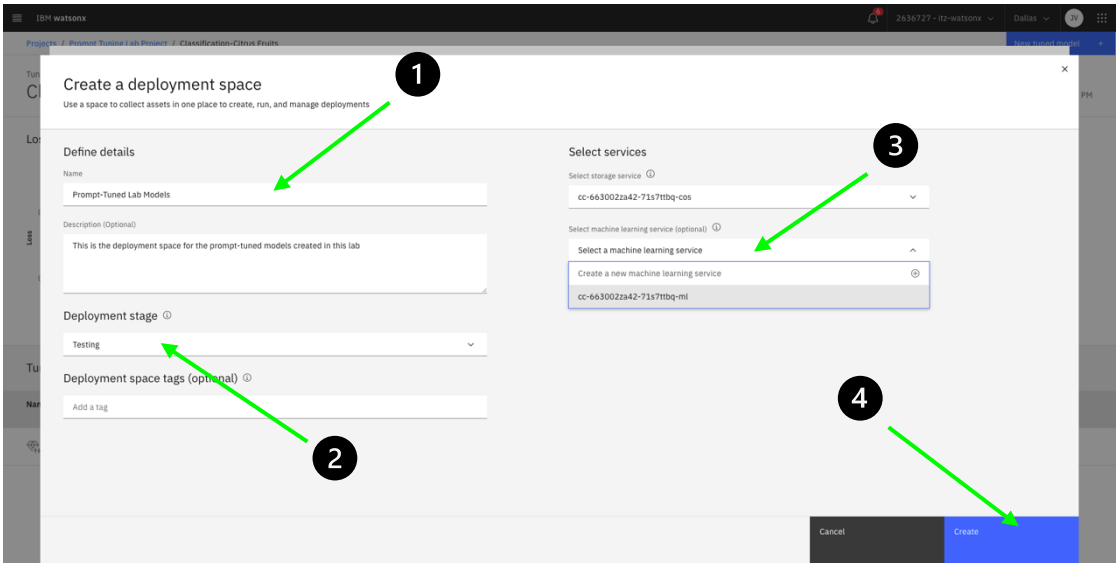
15. Model deployment is easy. Just hit the **New Deployment** button.



16. Define the details for your model deployment by giving it a **name**, *optional* **description**. Next, create a deployment space, by typing a name for your new space – continue to step 17.

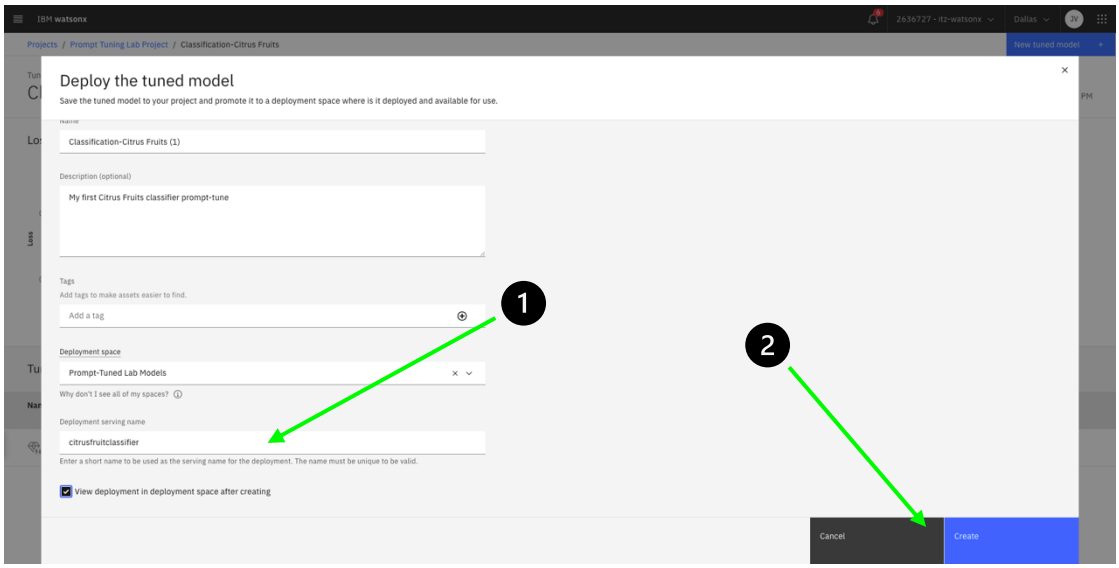


17. Define the details for your deployment space, such as **name**, *optional* **description**. Select the **Deployment Stage** from the drop-down. Select the associated machine learning service. Then click **Create.**

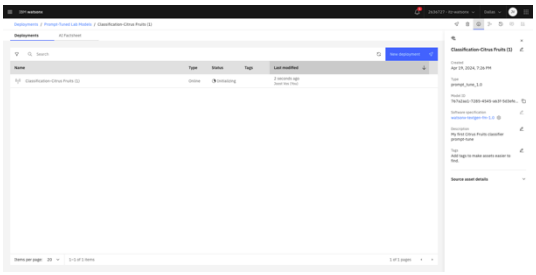


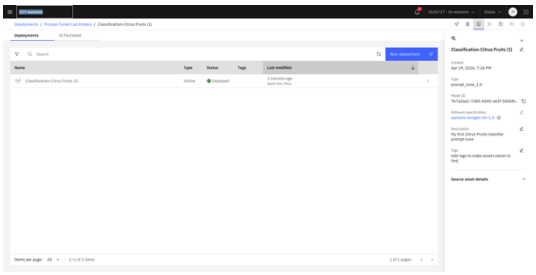
The space will be created. You are notified about the progress of this process.

18. You are now ready to deploy your prompt-tuned model. Give your prompt-tuned model a **Deployment serving name**. Then, click the **Create** button.

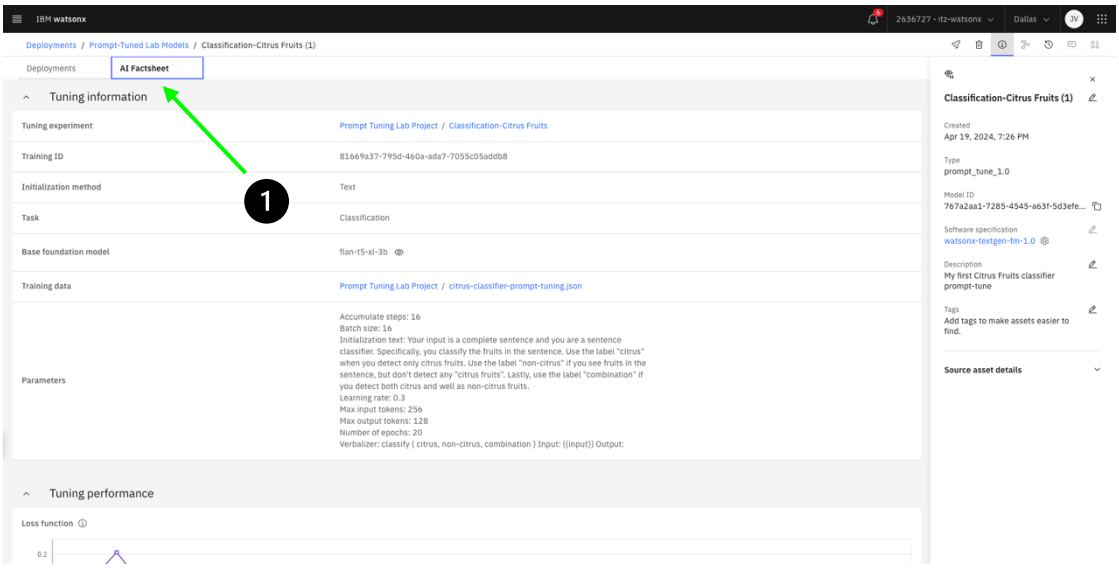


19. It takes some time before your model is deployed and available within the Prompt Lab. Grab a cup of coffee in the meantime!

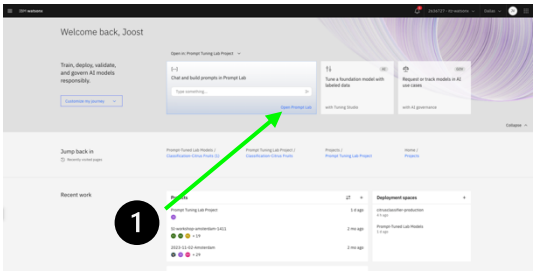


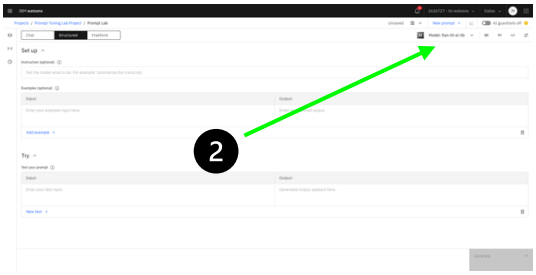


Once deployed, your prompt-tuned model is available in the Prompt Lab. Before heading over to the Prompt Lab, please take not of the AI Factsheet tab. When you are going to use the **watsonx.governance** for AI governance and AI life cycle management, its this AI Factsheet that forms part of your AI model inventory.

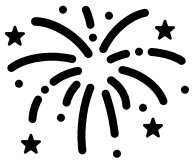


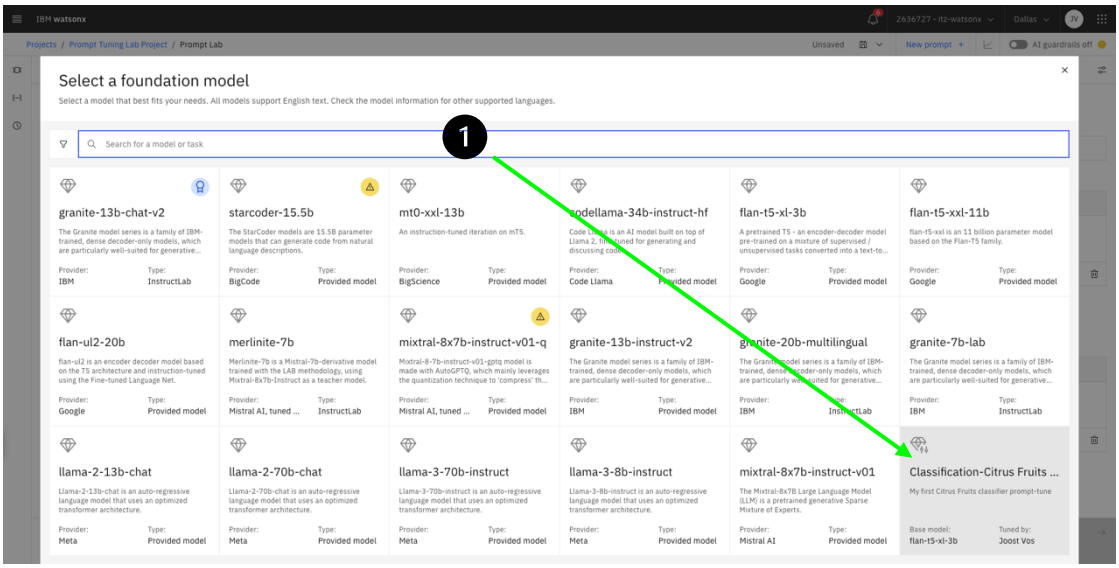
20. Let's go back to the Prompt Lab. Click on the **hamburger icon** on the top left in watsonx.ai. Then, click on Home, and then click **Open Prompt Lab.** Then, hover to **model drop-down menu** and click on it.



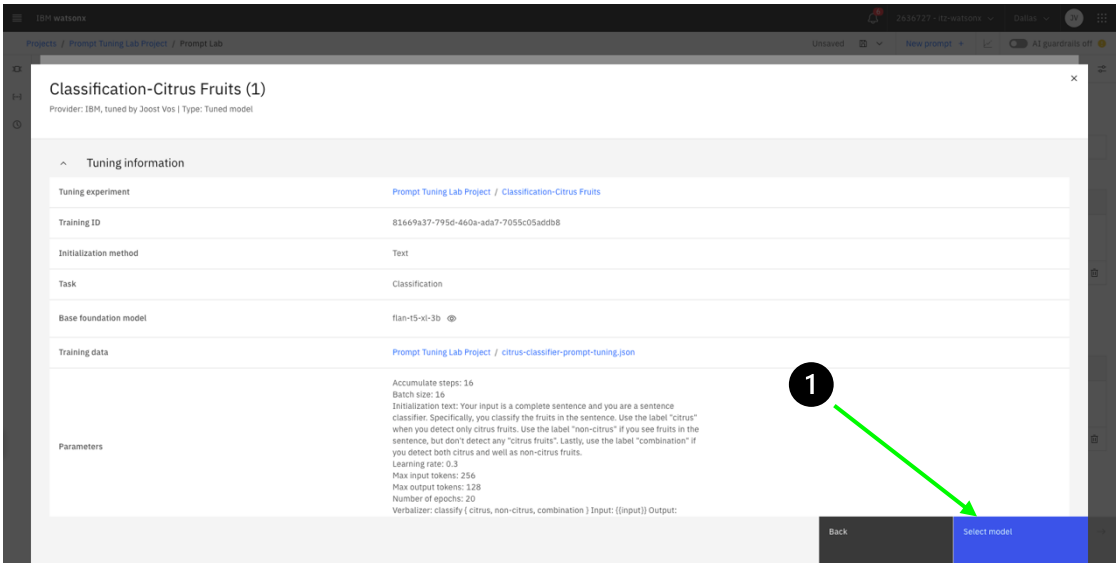


21. From the drop-down menu, click on **View all foundation models**. And there it is! ***Brilliant!*** Your first prompt-tuned model is available.



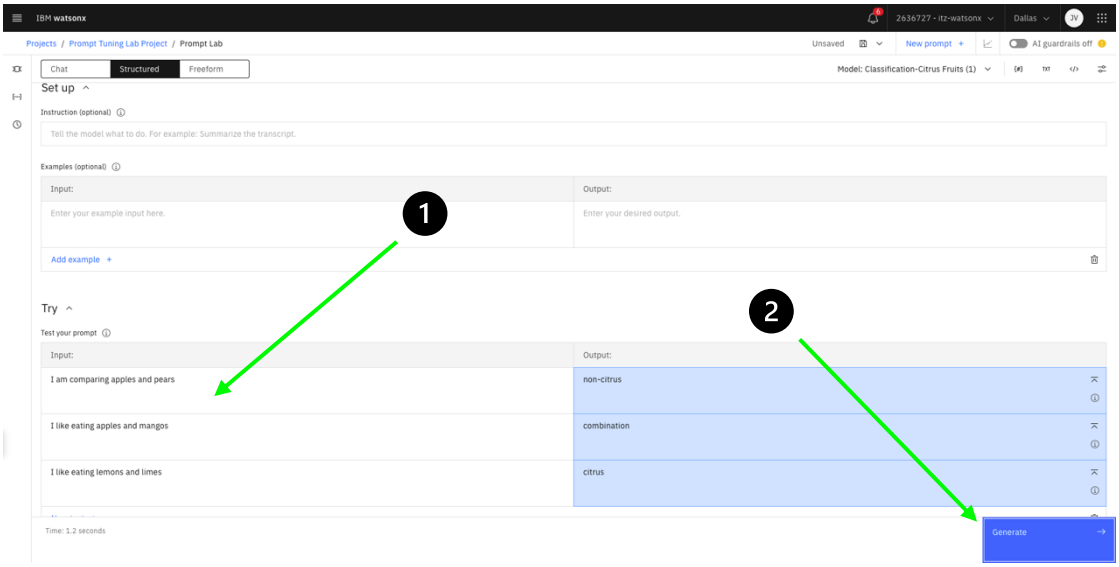


22. Select your prompt-tuned model and go for a test drive by clicking on **Select model.**



23. Be sure you are in the Prompt Studio's Structured mode. Use the next set up:
Instruction: *<not needed; you provided the instruction during prompt tuning>*
Input 1: *I am comparing apples and pears.*
Input 2: *I like eating apples and mangos.*
Input 3: *I like eating lemons and limes.*

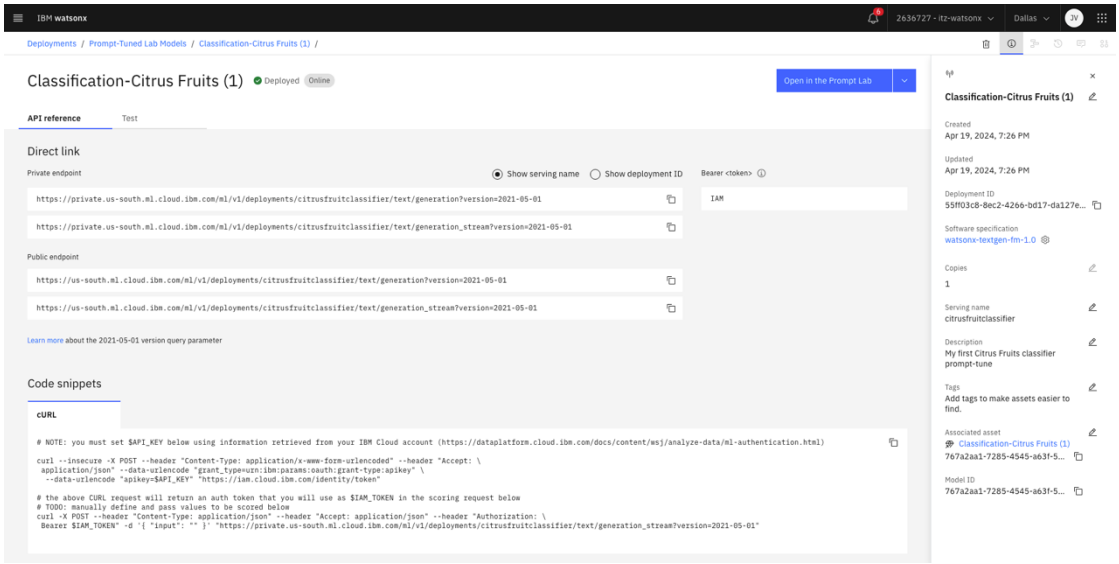
Test the following three inputs by adding them one by one to your input fields. You can add additional tests by hitting the **New test +** button. Once added, hit the **Generate** button and evaluate the results. If your tune was successful, you should see a screen like below:



24. All three inputs were classified by the prompt-tuned model. Feel free to test a couple of more examples.

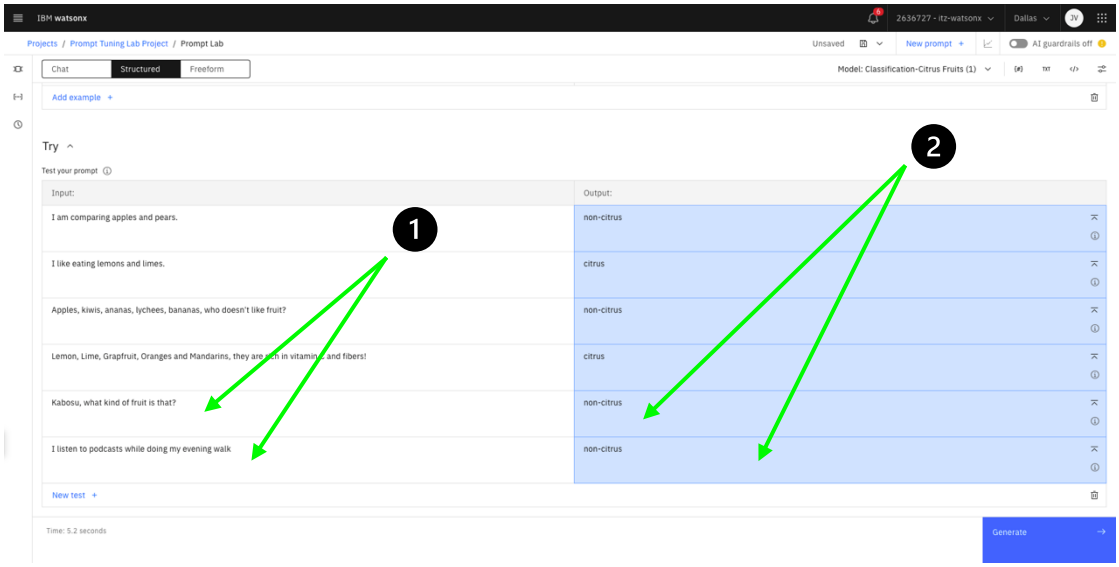
    Note: *although the performance of the prompt-tuned model is enhanced, it might misclassify input as we only used the bare minimum of 50 examples required to perform a prompt-tune*

25. Lastly, to find the **API reference**, including code snippets needed for integration, go to **hamburger menu** (top left) > **Deployments** > Select your **Model Deployment Space** > **Deployments** tab, and then click the **Name.** You should be getting a similar view as below**.** Here you can find everything needed for inferencing the prompt-tuned model from any application



Questions and Critical Thinking!

Continuing the case: You show your prompt-tuned model and its performance. Instantly, the marketing director is excited and urges you to push the classifier into production immediately. After a holiday trip to Japan, some major flaws come to the attention of the marketing director. Two critical issues were found. One of the issues is that a substantial portion of monitored posts are incorrectly classified. They don't mention fruits at all and get an incorrect label. Secondly, after visiting Japan, the marketing director got introduced to the citrus fruit **kabosu**. This one is not picked up by the classifier. The marketing director says to remember one sentence from one of your pitch presentations: "*garbage-in = garbage-out*". This is what was found:



Question 1: ***Did anything go wrong*** while prompt-tuning the **flan-t5-xl-3b** LLM? And if so, what went wrong? Or is this regular behavior of prompt tuned LLMs?

Question 2: Now you have a closer look at the step 23. You notice the misclassification of the input. **Can you explain why the mangoes is misclassified?**

Question 3: Can you improve, and if so, ***how would you improve*** the prompt-tuned classifier?

----------*End of Exercise 1*----------

3.4 Exercise 2 – Summarization.

Summarization Case: You have demonstrated your expertise in generative AI, your problem-solving skills, enthusiasm and craftmanship. You successfully solved the issues with the classification task. **LemonFresh** complimented you on your great job. They now want to continue with full speed ahead. By analyzing the posts on citrus fruits, they revealed some interesting insights into the interests of fruit lovers. Fruit lovers tend to post about ***specific health benefits*** of fruits. As **LemonFresh** is an online retailer of fruits, they are not nutrition experts with extensive knowledge on the health benefits of consuming citrus fruits. They want to promote their online retailing on FruitBowl. The platform allows for paid advertising. By tapping into the health interests, **LemonFresh** aims to reach FruitBowl users with relevant promotional information. To craft highly relevant messages, they first need to gain insights into the health benefits of oranges, clementines and grapefruits – their best-selling citrus fruits. As a first step, they found an insanely good online resource describing some of the citrus fruit's health benefits. Have a look yourself! You can find the page here:

https://insanelygoodrecipes.com/citrus-fruits/

LemonFresh looked at the website They are particularly interested in the following health effects:

Supports the immune system	Regulates blood pressure levels
Rich in antioxidants	Provides skin care benefits
Contains essential nutrients	Improves bone strength
Supports eye health	Reduces bad cholesterol
Relieves stress	Aids in digestion
Contains fiber	High in vitamin C

LemonFresh aims to provide relevant messages based on scientific evidence. So, they investigated PubMed – the world's largest literature database on biomedical and medical research. They did this online search on PubMed: (click the link to get there)

https://pubmed.ncbi.nlm.nih.gov/?term=health+benefits+citrus+fruits+AND+%28orange+OR+lemon+OR+lime+OR+grapefruit+OR+tangerine+OR+mandarin%29+NOT+drugs&filter=simsearch1.fha&format=abstract

It's too much data! They've asked you as IBM business partner to summarize these abstracts. Your task is to extract the fruit its health benefits in just 2-3 sentences from each abstract. ***Off we go!***

Prompt-tuning – the summarizer

You can work in the same project as for exercise 1.

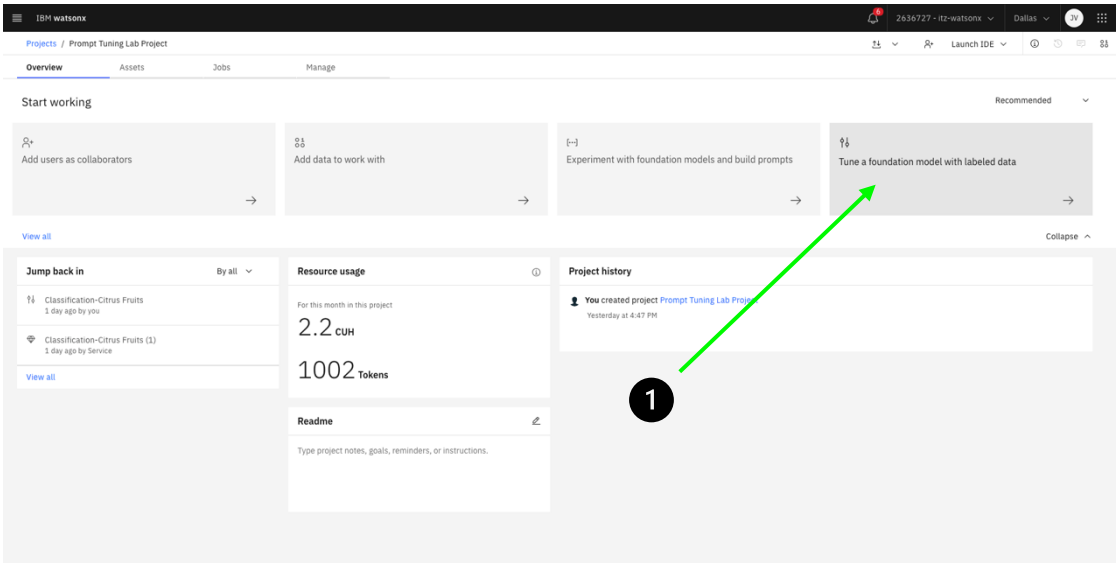
1. First go to the **Prompt Lab** for some model testing for the summarization task of several models. At this step, you can pick any model you think might produce good summaries.

2. Go to your **Project page** in watsonx.ai (**step 7; exercise 1; page 17**)

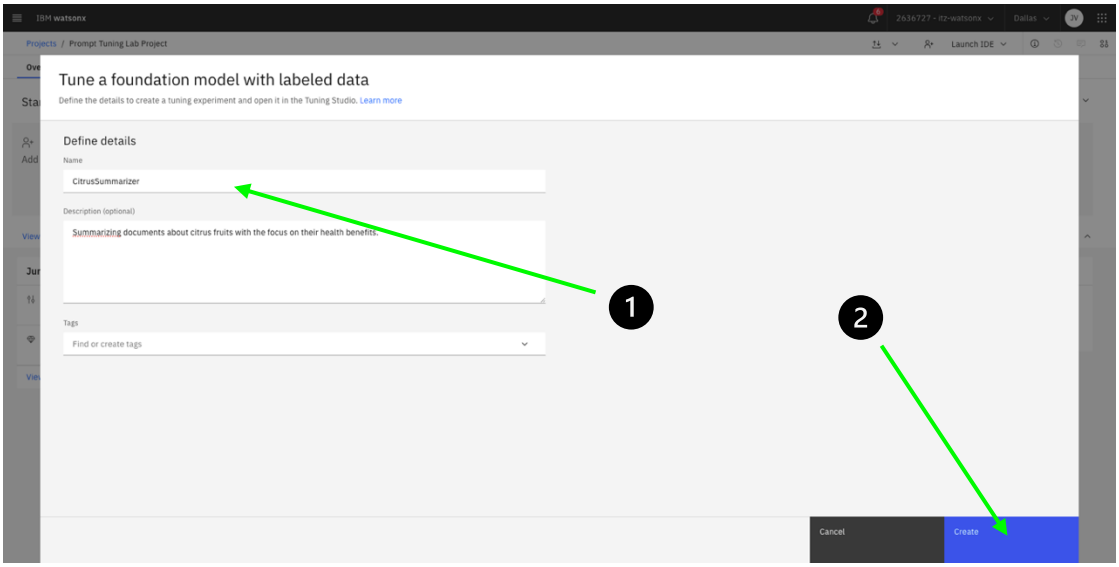
3. For your summarization prompt-tune you need to complete a pre-existing JSON that we have created for you. This JSON only contains 50 input documents, but only 25 summaries. You can download a **JSON tuning file** by clicking the link below:
   https://github.com/joost-vos/watsonx-ai-prompt-tuning/blob/main/prompt-tuning-data/50_abstract_summaries_granite-13b-instruct-v2_students_version.json

4. It is up to you how to create the other 25 summaries. Tip: *use the prompt lab.* **When you are done, you are ready to start your prompt tune for the summarization task.**

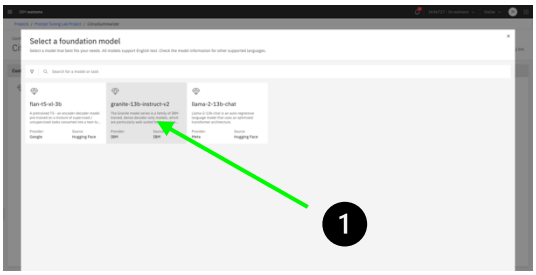
5. Click on the "Tune a foundation model with labeled data"

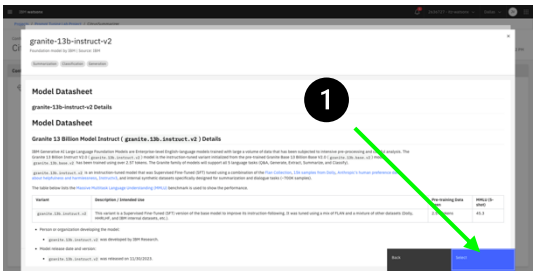


6. Give the tune a name and *optional* description. Next, click **Create.**

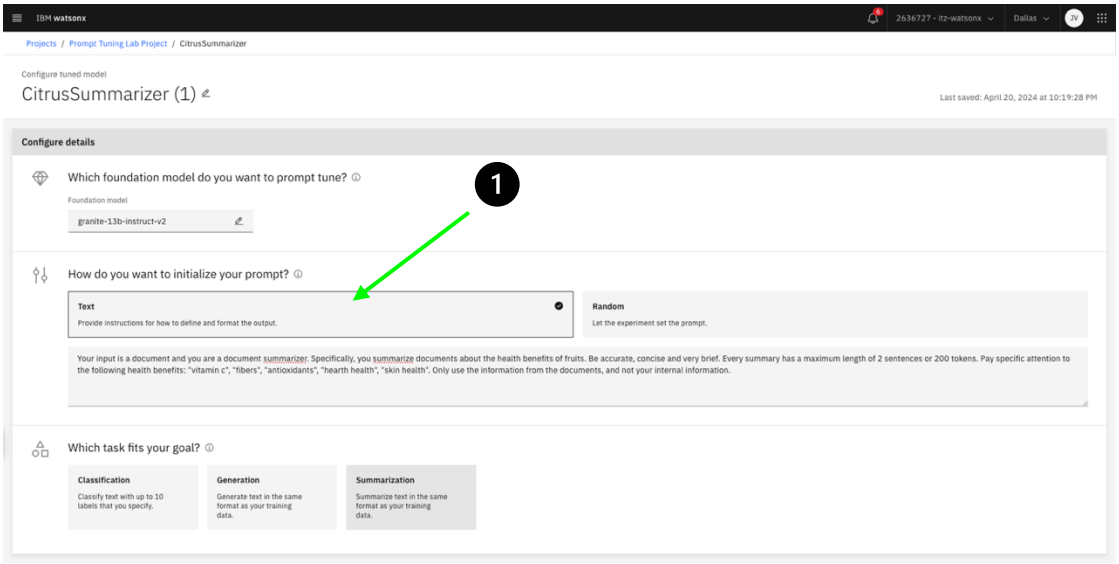


7. Instead of working with the flan-t5-xl-3b model you choose to work with IBMs **granite-13b-instruct-v2 model**, because you've learned from the model card that this model works well with academic literature. You select **granite-13b-instruct-v2:**



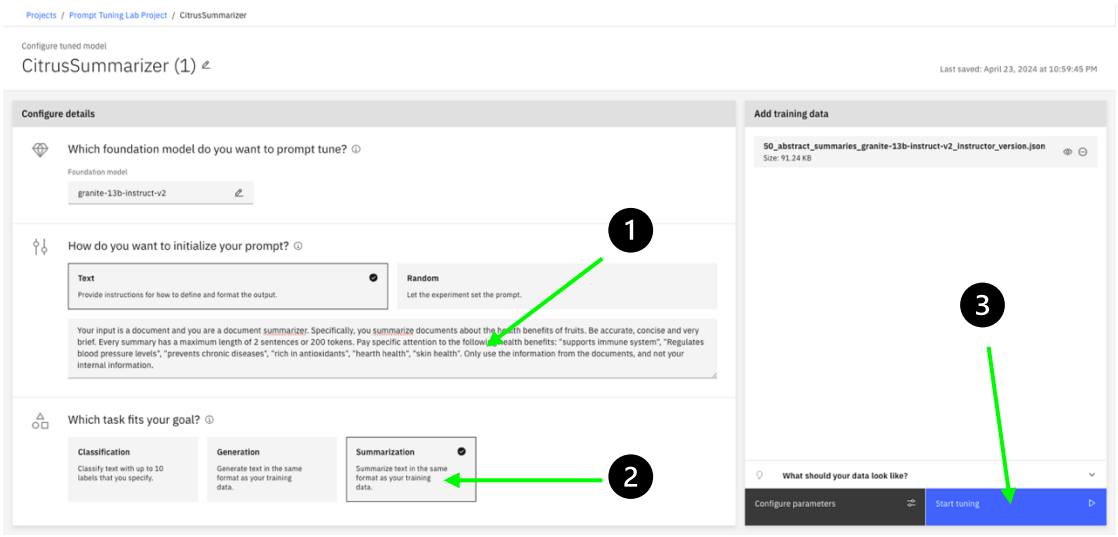


8. You start with the tune by selecting **Text** as way to initialize your prompt.



9. Then give your instruction – feel free to play with this – In step 4 you probably played with this. As a starter you can use the following, and click **Start Tune:**

Your input is a document, and you are a document summarizer. Specifically, you summarize documents about the health benefits of fruits. Be accurate, concise, and very brief. Every summary has a maximum length of 2 sentences or 200 tokens. Pay specific attention to the following health benefits: "supports immune system", "Regulates blood pressure levels", "prevents chronic diseases", "rich in antioxidants", "hearth health", "skin health". Only use the information from the documents, and not your internal information.



10. After some minutes this task is completed. Here's the summary of your tuning experiment:

11. You are somewhat surprised by the limited decrease in loss function. Let's evaluate the prompt-tuned model in the Prompt Lab.

12. You deploy the model similarly as you did before (step 15-19 of exercise 1)

13. Test drive your prompt-tuned model using the following two abstracts. Click on the links below the images. Navigate to PubMed and copy & paste the abstract text into the Prompt Lab try panel:

> Pharmaceutics. 2021 Oct 31;13(11):1818. doi: 10.3390/pharmaceutics13111818.

Citrus Flavanone Narirutin, In Vitro and In Silico Mechanistic Antidiabetic Potential

Ashraf Ahmed Qurtam [1], Hamza Mechchate [2], Imane Es-Safi [2], Mohammed Al-Zharani [1], Fahd A Nasr [3], Omar M Noman [3], Mohammed Aleissa [1], Hamada Imtara [4], Abdulmalik M Aleissa [5], Mohamed Bouhrim [6], Ali S Alqahtani [3]

Affiliations + expand

PMID: 34834233 PMCID: PMC8619962 DOI: 10.3390/pharmaceutics13111818

Abstract

Citrus fruits and juices have been studied extensively for their potential involvement in the prevention of various diseases. Flavanones, the characteristic polyphenols of citrus species, are the primarily compounds responsible for these studied health benefits. Using in silico and in vitro methods, we are exploring the possible antidiabetic action of narirutin, a flavanone family member. The goal of the in silico research was to anticipate how narirutin would interact with eight distinct receptors implicated in diabetes control and complications, namely, dipeptidyl-peptidase 4 (DPP4), protein tyrosine phosphatase 1B (PTP1B), free fatty acid receptor 1 (FFAR1), aldose reductase (AldR), glycogen phosphorylase (GP), alpha-amylase (AAM), peroxisome proliferator-activated receptor gamma (PPAR-γ), alpha-glucosidase (AGL), while the in vitro study looked into narirutin's possible inhibitory impact on alpha-amylase and alpha-glucosidase. The results indicate that the studied citrus flavanone interacted remarkably with most of the receptors and had an excellent inhibitory activity during the in vitro tests suggesting its potent role among the different constituent of the citrus compounds in the management of diabetes and also its complications.

Keywords: enzyme; isonaringin; mechanism of action; molecular docking; naringenin rutinoside; narirutin; receptors.

Conflict of interest statement

The authors declare no conflict of interest.

Cited by 11 articles | 68 references | 10 figures

Link 1:
https://pubmed.ncbi.nlm.nih.gov/34834233/

> Foods. 2023 Sep 18;12(18):3469. doi: 10.3390/foods12183469.

Citrus Carotenoid Extracts Exert Anticancer Effects through Anti-Proliferation, Oxidative Stress, and Mitochondrial-Dependent Apoptosis in MCF-7 Cells

Juanjuan Wei [1], Yurong Li [1], Zimao Ye [1], Yi Li [2], Zhiqin Zhou [1, 3]

Affiliations + expand

PMID: 37761178 PMCID: PMC10529845 DOI: 10.3390/foods12183469

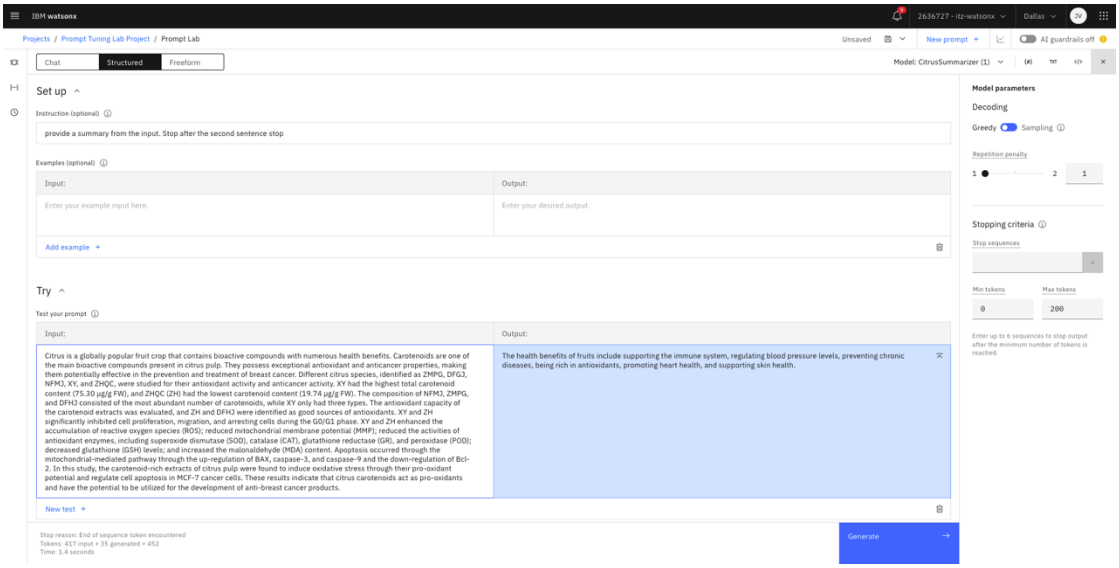
Abstract

Citrus is a globally popular fruit crop that contains bioactive compounds with numerous health benefits. Carotenoids are one of the main bioactive compounds present in citrus pulp. They possess exceptional antioxidant and anticancer properties, making them potentially effective in the prevention and treatment of breast cancer. Different citrus species, identified as ZMPG, DFGJ, NFMJ, XY, and ZHQC, were studied for their antioxidant activity and anticancer activity. XY had the highest total carotenoid content (75.30 μg/g FW), and ZHQC (ZH) had the lowest carotenoid content (19.74 μg/g FW). The composition of NFMJ, ZMPG, and DFHJ consisted of the most abundant number of carotenoids, while XY only had three types. The antioxidant capacity of the carotenoid extracts was evaluated, and ZH and DFHJ were identified as good sources of antioxidants. XY and ZH significantly inhibited cell proliferation, migration, and arresting cells during the G0/G1 phase. XY and ZH enhanced the accumulation of reactive oxygen species (ROS); reduced mitochondrial membrane potential (MMP); reduced the activities of antioxidant enzymes, including superoxide dismutase (SOD), catalase (CAT), glutathione reductase (GR), and peroxidase (POD); decreased glutathione (GSH) levels; and increased the malonaldehyde (MDA) content. Apoptosis occurred through the mitochondrial-mediated pathway through the up-regulation of BAX, caspase-3, and caspase-9 and the down-regulation of Bcl-2. In this study, the carotenoid-rich extracts of citrus pulp were found to induce oxidative stress through their pro-oxidant potential and regulate cell apoptosis in MCF-7 cancer cells. These results indicate that citrus carotenoids act as pro-oxidants and have the potential to be utilized for the development of anti-breast cancer products.

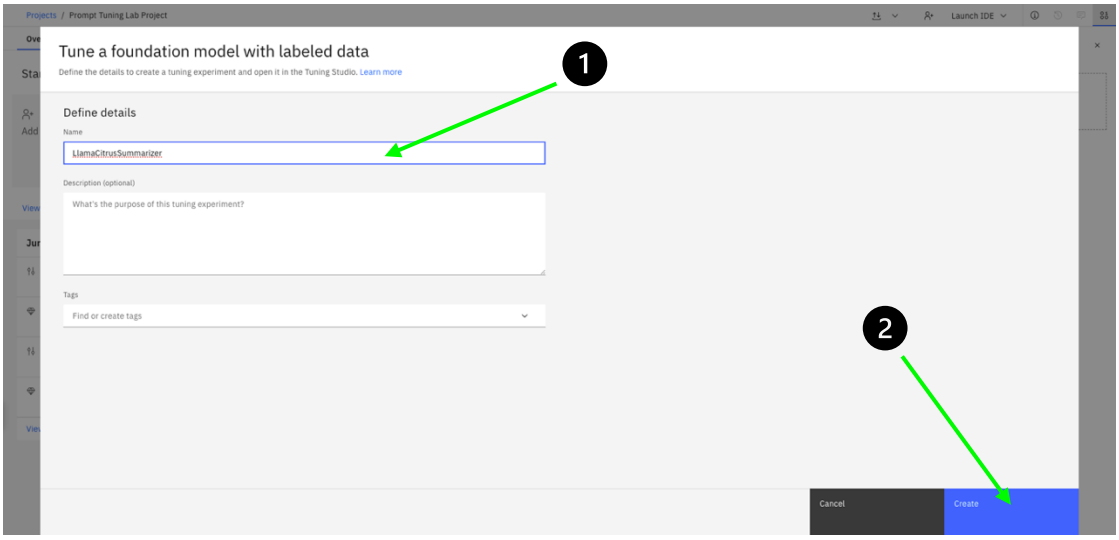
Keywords: apoptosis; carotenoids; citrus; oxidative stress; reactive oxygen species (ROS).

Link 2:
https://pubmed.ncbi.nlm.nih.gov/37761178/

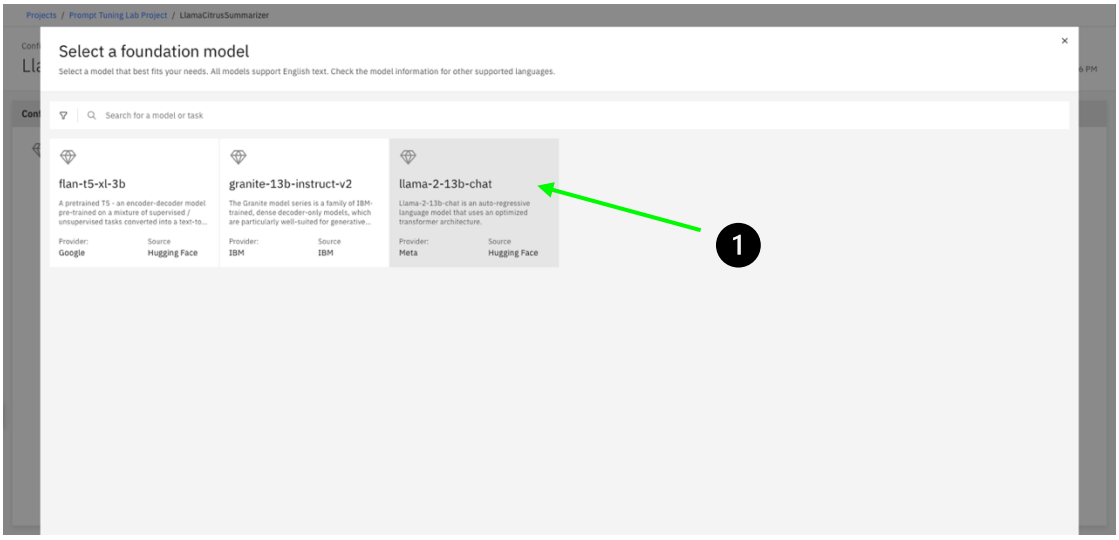
14. The following screenshot is an example output of your prompt-tuned **granite-13b-instruct-v2** model. It looks decent, despite your initial doubts caused by the limited decrease in loss function.



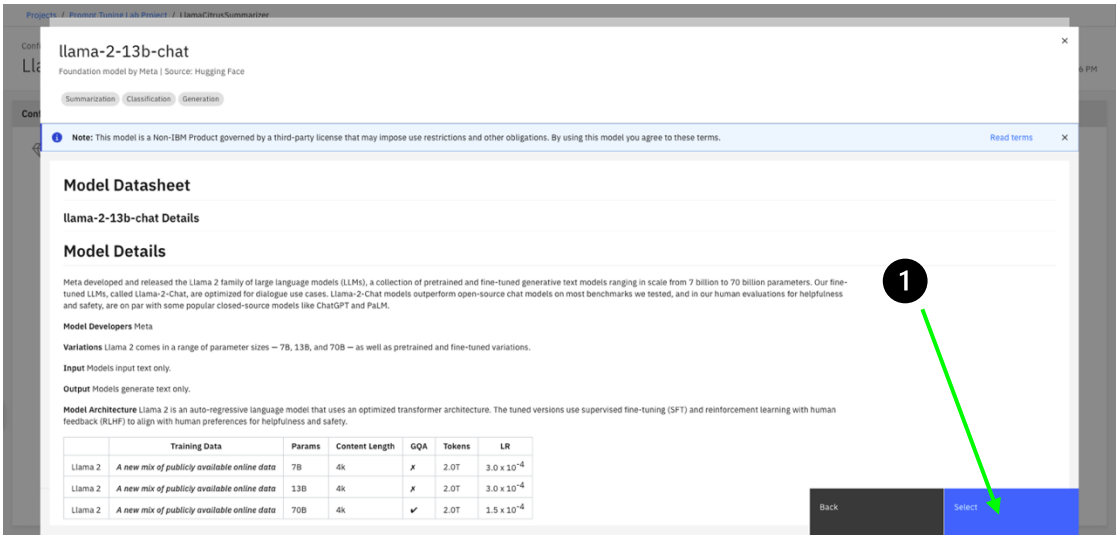
15. You now want to benchmark the summarization capabilities between different models. You decide to prompt-tune the third model in watsonx.ai that is available for prompt-tuning, llama-2-13b-chat. **Repeat the steps 5-12 from exercise 2** and make sure you select **llama-2-13b-chat** as foundation model.



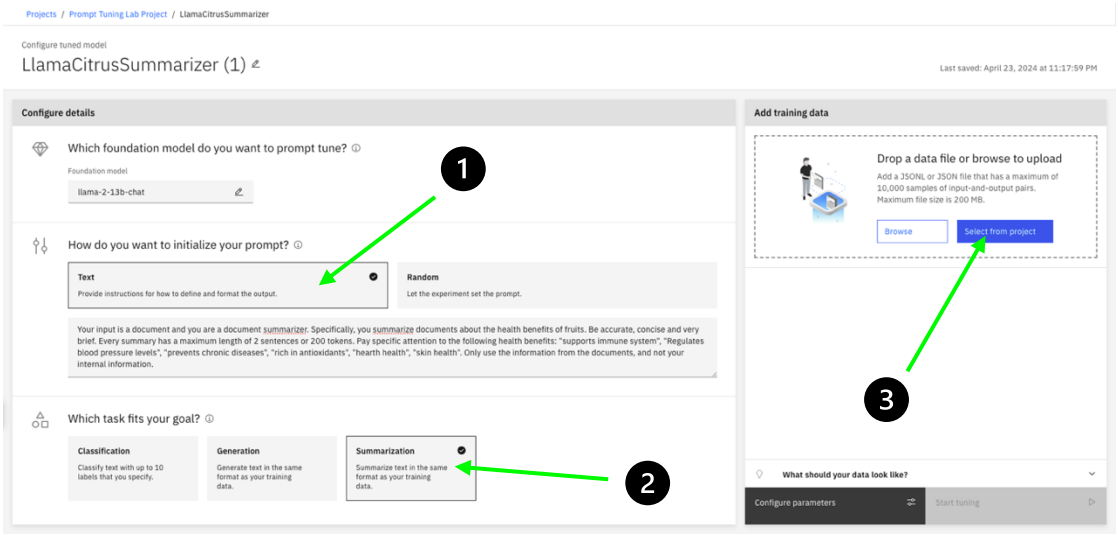
16. Select the Llama-2 foundation model and click **Create.**



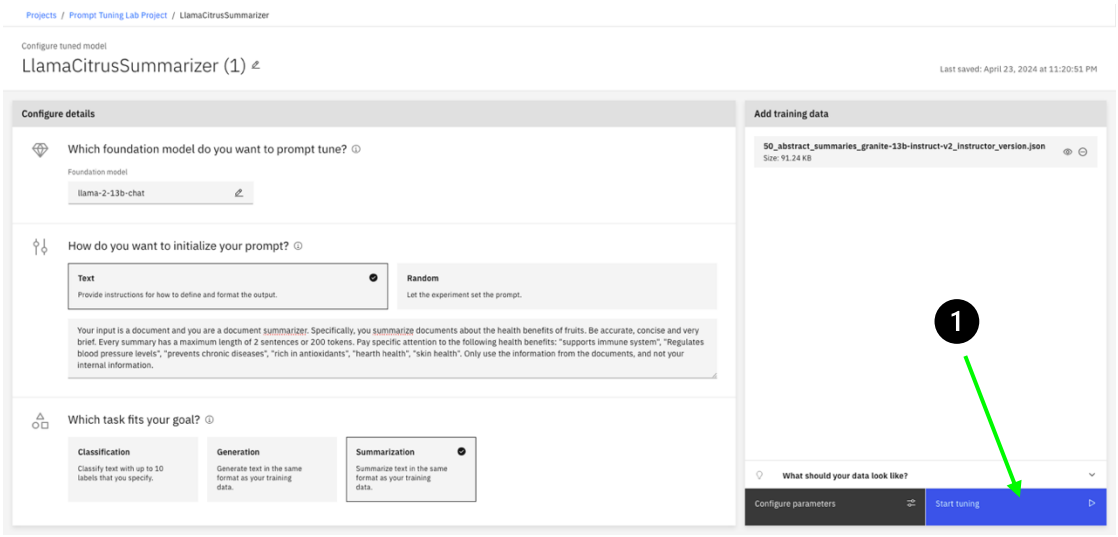
17. Have a look at the model card.



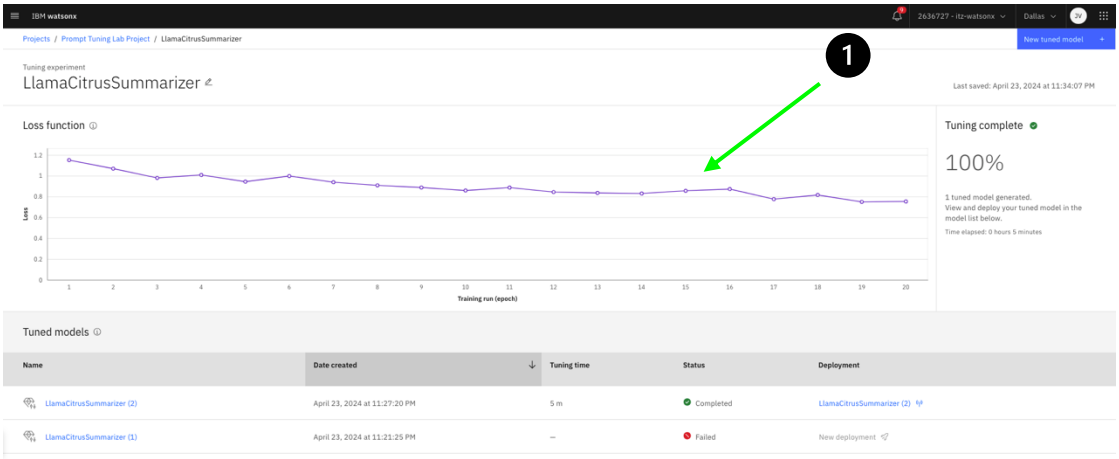
18. Fill out the configuration details as before – don't upload the tuning training data again! Select the existing project asset!



19. You select the existing 50_abstract_summaries_... asset and click **Select asset** and in the next screen click **Start tuning.**

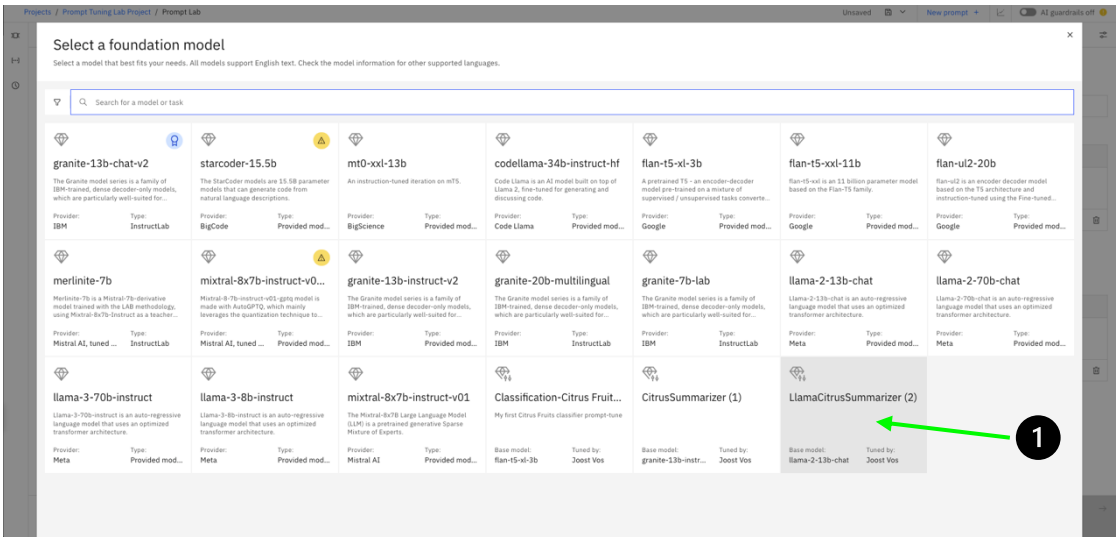


20. When done training, evaluate the decline in Loss Function. Albeit slightly, with the prompt-tune, the **llama-2-13b-chat** model improved in summarizing documents describing the citrus fruits health benefits over its base capabilities.



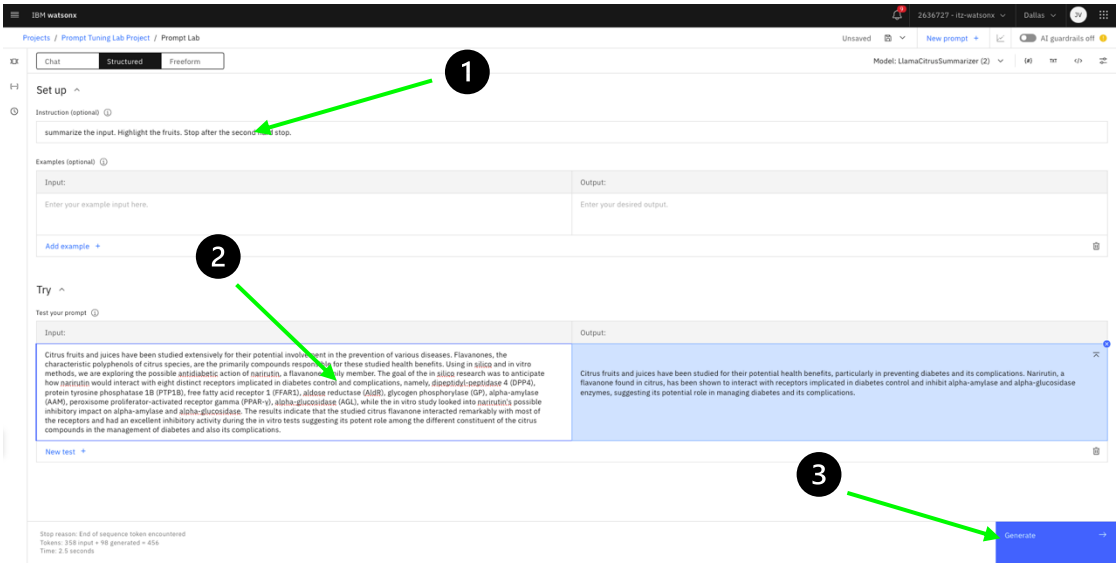
21. You decide to promote this tuned model to the deployment space for a test run in the prompt lab. You deploy the model similarly as you did before (step 15-19 of exercise 1).

22. Go the Prompt Lab and test driving the prompt-tuned llama-2-13b-chat model as described on page 32 with the 2 abstracts mentioned in step 13.



23. Test your Llama prompt-tuned model in Prompt Lab. You may need to provide an instruction. As instruction you can test the following one:

"summarize the input. Highlight the fruits. Stop after the second hard stop."



24. Now, alternate between the two prompt-tuned models you have created.

25. What are your thoughts?

Questions and Critical Thinking!

Continuing the case: You show your summarization prompt-tuned models to your client. The marketing director sees good progress. Both models demonstrate decent summarization performance based on the limited training data that was fed to the prompt-tuning process. The marketing director has a slight preference for IBMs granite model. Please read the following questions.

Question 4: Is it possible to improve its summarization capabilities tailored to scientific papers describing the health benefits of citrus fruits. How would you approach that?

Question 5: the IBM model seem to respond stricter to its instructions. Can you explain as to why this is?

----------*End of Exercise 2*----------

3.5 Lab 3 – Generation.

Generation Case: You've come a long way and made some serious progress with customizing models specifically for your client. LemonFresh applauds you on the excellent work. You've been able to extract the relevant posts from the FruitBowl platform mentioning citrus fruit consumption. You've been able to extract relevant and insights from the scientific literature. With these ingredients, you can craft some sound and science-proven marketing content for their beloved FruitBowl audience. Now it is showtime! You are about to start generating marketing posts about ***specific health benefits*** tailored and targeted at your audience at FruitBowl. LemonFresh has not decided on the specific tone-of-voice. Nor do they want to submit paid message fully automatically without human review. They value the *'human-in-the-loop'* principle.

In this last step, LemonFresh wants you to propose which model to select from available models in watsonx.ai. They've heard about the latest model release by Meta. Question is, which of the models would be up to the task of writing appealing targeted message that reflect LemonFresh's wishes, that is creating engaging and relevant messages sharing potential health benefits derived from scientific literature. Secondly, they ask you to demo your most engaging and powerful post suggestion. Lastly, they want to receive an approach/proposal of how to create training data to prompt-tune a model, without the need of having to prompt-engineer every time you propose the generation of a post suggestion.

They want you to:

- -Compare various models in marketing message generation
- -Develop a prompt that delivers engaging and powerful marketing messages
- -Develop an approach/strategy to be able to prompt-tune for the generation task (...but don't do the prompt-tune itself!)

They've asked you as IBM business partner to assist LemonFresh in this endeavor. It's play time! ***Off we go!***

Exercise 3 – the generator

You can work in the same project as for exercise 1. With the exercises 1 and 2 you have seen the essential steps and methods to tackle exercise 3. In exercise 3, we encourage you to ***think*** about how to develop and design an approach that creates value to your client. We intentionally did not provide guidance. Use your commonsense reasoning skills. Iterate, evaluate, use your imagination and experiment….and have some fun alongside!

The only concrete pointers we give is:

1. With watsonx.ai you have all the tools at hand to train, tune, validate and deploy language models.
2. Start simple and small
3. Pull some ideas and inspiration from exercise 1 and 2.
4. Use the output from the summarization prompt-tuned model of choice as input for the generation ;-)
5. Have fun!

As a teaser, we provide some examples of what might be possible as output:

Details	Post
Citrus fruits, such as oranges, lemons, and grapefruits, are rich in bioactive compounds that have numerous health benefits, including antioxidant and anticancer properties. The carotenoids present in citrus pulp have been found to induce oxidative stress and regulate cell apoptosis in MCF-7 cancer cells...	"Boost Your Health This Summer with Our Juicy Fruits 🍊🍎 At #FruitBowl , we're celebrating healthy eating habits by sharing our love 💕for juicy & nutritious citrus fruits like Oranges, Lemons & Grapefruits! These yummy treats aren't just refreshing but also packed with antioxidants & anticancer properties 😋 ...
Eriocitrin (ERI), a natural flavonoid found in citrus fruits, has been shown to have antioxidant, anti-inflammatory, anti-diabetic, and anti-tumor properties. ERI effectively alleviated LPS-induced pathological injury in mice by suppressing the inflammatory response and decreasing oxidative stress in...	"Get ready to zing up your summer with Vitamin C-rich citurs fruis 🍹👯 Did you know consuming foods high in vitamins can help boost immunity against diseases, making them perfect additionsto yoursun-kissed days ahead?! ❤ Buy now from @LemonFresh"
Grapefruit contains bioactive furanocoumarins that contribute to its health benefits, including antioxidant, anti-inflammatory, and anticancer activities. These compounds modulate molecular pathways, such as the signal transducer and activator of transcription 3, nuclear factor-κB, ...	"Summer is here! ☀️ Beat the heat with refreshing Grapefruits 🍊 Rich in antioxidants, anti-inflammatory & anticancer properties, this superfood will keep you cool & protected all season long! 💪 Get your daily dose of goodness from @LemonFresh today!"

Disclaimer: The labs are designed for ***educational purposes*** only. They are not intended or designed to provide health or healthy lifestyle advice in what form. If you need health advice or tips how to improve your healthy lifestyle, consult your medical professional.

Appendix

Appendix 1 - Answers to the exercise questions

Question 1:

Nothing went wrong, and it is not regular behavior of an LLM. First, the misclassification of sentences not mentioning fruit at all. We didn't include a label for 'negative' sentences in which none of our labels is present. An LLM always gives an answer, even if we don't want it to. That's what you are looking at. Second, the missing of citrus fruits by the classifier. With just 50 examples we just used too little training data for the prompt-tune. Check the training data. You will soon notice that many citrus fruits are not there in the training data, certainly not the lesser-known citrus fruits. Lastly, always test rigorously before putting a model into production 👍

Question 2:

Have a look at the training data. It is limited in size. The fruit mango occurs twice in a sentence that was labeled with 'combination' and just once in a sentence labeled as 'non-citrus'. Aside from the limited amount of training data, the number of provided examples is disbalanced. It's statistics – an LLM will provide the most probable answer – in this case, 'combination' is the most probable one. Solution to this problem is to bring balance into the training data. Give equal amounts of examples for each of the representative labels.

Question 3:

Yes, you can improve. To solve the issue of misclassification of sentences not mentioning fruit at all, do introduce a label like "no fruits mentioned". Include evenly distributed relevant training data with similar amounts of training data per label. To solve the issue of misclassification of lesser-known citrus fruits, extend your training data with a wider array of known citrus fruits. You can use online sources to extend the number of covered citrus fruits.

Question 4:

It happens that the out-of-the-box settings may not immediately lead to satisfying results. Tuning a foundation model in watsonx.ai is an iterative process. You run a tuning experiment and then evaluate the results. If necessary, you change experiment variables and rerun the experiment repeatedly until you are satisfied with the output from the tuned foundation model. Check your progress after each experiment run. It is a good practice to find any limitations in your tuning experiment configuration and address them before you assess your training data for potential problems. You can read more about evaluating tuning experiment results here:

https://dataplatform.cloud.ibm.com/docs/content/wsj/analyze-data/fm-tuning-methodology.html?context=wx&audience=wdp#edit-tuning-parameters

Appendix 2 – Learning Rate

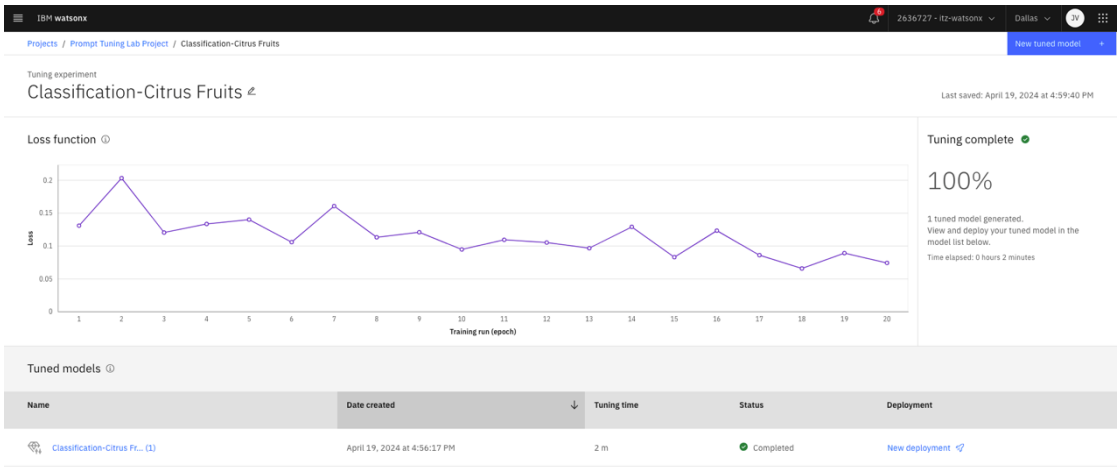
By Felix Lee, IBM

This is not meant to be a thorough discourse of Learning Rate, but sufficient information to understand why you may need to tune the Learning Rate parameter when you are performing prompt tuning.

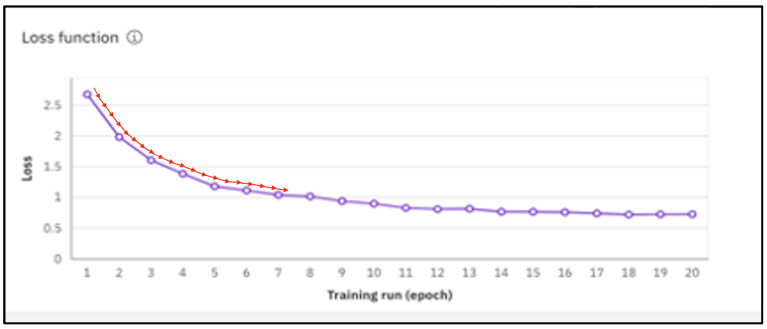
As seen in Step 14 of Exercise 1 and Step 10 and Step 20 of Exercise 2, part of prompt tuning is the calculation of the **Loss function.** At a simple level, the **Loss function** measures how well the LLM models your dataset. The closer the LLM's completion is to the actual desired output, the smaller the **Loss**. So, in prompt tuning, you want to find the lowest point of the **Loss function**. Ideally, one would like to see it reach zero – but that is difficult (impossible almost).

This is where **Number of epochs** and **Learning rate** come in. The **Number of epochs** is how many times to cycle through the training data. While in general the more epochs the better, there will be a point of diminishing return where the cost of another epoch brings no further improvement.

Here is the graph of the **Loss function** from Step 14 of Exercise 1:

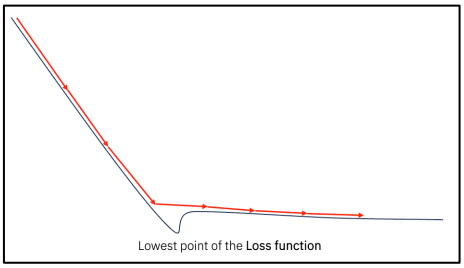


In this function, the lowest value seems to be around 0.074. The distance between each dot can be thought of as the **Learning rate**. If the **Learning rate** value is too low, then even after 20 epochs the **Loss function** might have only reached a minor value decrease. This means that the LLM is not close to providing the best completion for the test data.



One might be tempted to use a large **Learning rate**. This may not be a bad idea if the **Loss function** is shaped as above – and ever dipping line. A larger **Learning rate** may get to the lowest point sooner (or as low as the **Number of epochs** allows).

However, a high **Learning rate** means you are allowing the model algorithm to accept large changes. You can "overshoot" the lowest point of the **Loss function**. For example. If the actual **Loss function** looks like the black curve below.



The red arrows reflect a large **Learning rate,** so it progresses quickly. However, because of the large steps it completely missed the lowest point of the **Loss function**. This means that the algorithm change to the LLM will not be optimal, even though it may be "not bad".

The natural question arises of how to set the **Learning rate**. Here is a suggestion:

- Start with the default value.
- If the **Loss function** is levelling off early (as in the example in Section 6.2), try a smaller **Learning rate** to see if there might be hidden valleys.
- If the **Loss function** is still showing continued decline after 20 epochs, try a larger **Learning rate** or a higher **Number of epochs**.

Always keep in mind – the **Loss function** is calculated based on the data you put in. So, while you might get to a levelling off at a low value, it just means the LLM is now able to perform generations reflective of the labeled data set. It does NOT necessarily mean it is optimally tuned to your entire data set.

Appendix 3 - Summary

- **Prompt Tuning is a powerful tool to help train a model to perform specific tasks** (especially when there are business rules that are not known outside of the company).
- It is important to pay attention to the content of the training data. LLMs are quite smart, and **the labeled data input does not determine how the LLM will perform a completion, but it does help to steer it in a certain direction**. As such, the data used mustn't inadvertently introduce new biases to the model.
- Take care in creating the prompt tuning data set.
    - Spend time with the client to come up with a set of data that can augment the knowledge base of the model. This depends on the downstream tasks you need the LLM to perform.
    - The key is to highlight what the LLM does not naturally do (such as identifying multiple categories in a classification task), or rules (such as business rules) that it could not have learned with its training.

      For example, in Exercise 1, you want to steer the LLM toward understanding the differences between citrus fruits and other fruits. Your prompt tuning data set should include plenty of examples for each of the category labels. Ideally, the number of samples per label should be equal. Simply because an imbalance may lead to bias of the output towards the most probable output, instead of the most likely correct output.
    - In contrast, a straight percentage representation of the actual data may not be useful. If you have 1 million messages in your database and 900,000 of those are about citrus fruits, then a % representation means you will create a 200-sample dataset with 180 entries with a citrus label output, and only 20 entries with one of the other labels. This may be valid. At the same time, it will steer the LLM towards a high probability of providing the citrus label output. This would have caused defects in the purpose of the prompt tuning.