Getting Started with   
SAP Leonardo Machine Learning Foundation

WDEML1 – Collection 2018/Q3

Exercises / Solutions  
SAP SE

Table of ContentS

[Authorization 3](#_Toc528077746)

[0 Try the SAP Leonardo Demo Page 4](#_Toc528077747)

[0.1 Try out the Person Finder 4](#_Toc528077748)

[0.2 Try out the further demo scenarios 7](#_Toc528077749)

[1 Ready-to-use Services via API Business Hub 8](#_Toc528077750)

[1.1 Use SAP Leonardo ML Document Feature Extraction 8](#_Toc528077751)

[1.2 Use SAP Leonardo ML Similarity Scoring 12](#_Toc528077752)

[1.3 Use SAP Leonardo ML Topic Detection 16](#_Toc528077753)

[1.4 Use SAP Leonardo ML Image Processing APIs 19](#_Toc528077754)

[1.4.1 Use SAP Leonardo ML Image Classifier Service 19](#_Toc528077755)

[1.4.2 Use SAP Leonardo ML Image Feature Extraction API 19](#_Toc528077756)

[1.4.3 Use SAP Leonardo ML Optical Character Recognition 20](#_Toc528077757)

[2 Ready-to-use Services in application development 21](#_Toc528077758)

[2.1 Setup Development Environment in SAP Cloud Platform 21](#_Toc528077759)

[2.2 Use SAP Leonardo ML Image Classification with SAPUI5 24](#_Toc528077760)

[3 SAP ML Foundation in SAP Cloud Foundry 32](#_Toc528077761)

[3.1 Install Cloud Foundry Command Line Interface (CF CLI) 32](#_Toc528077762)

[3.2 Create ML Foundation Service Instance & Service Key 32](#_Toc528077763)

[3.2.1 Create Service Instance and Service Key for ML Foundation Beta (optional) 37](#_Toc528077764)

[4 Customizable Services 38](#_Toc528077765)

[4.1 Generate Access Token 38](#_Toc528077766)

[4.2 Customizable Image Classification 39](#_Toc528077767)

[4.2.1 Deploy Re-training Application 40](#_Toc528077768)

[4.2.2 Training data 42](#_Toc528077769)

[4.2.3 Re-train the Image Classifier Model 43](#_Toc528077770)

[4.2.4 Deploy the re-trained Image Classifier Model 45](#_Toc528077771)

[4.3 Customizable Text Classification 48](#_Toc528077772)

[4.3.1 Re-train the Text Classifier Model 49](#_Toc528077773)

[4.3.2 Deploy the re-trained Text Classifier Model 52](#_Toc528077774)

[4.3.3 Deploy test app & test re-trained Text Classifier Model 54](#_Toc528077775)

[4.4 Customizable Image Object Detection 58](#_Toc528077776)

[4.4.1 Training Data (optional) 59](#_Toc528077777)

[4.4.2 Re-train the Image Object Detection Model 60](#_Toc528077778)

[4.4.3 Deploy the re-trained Object Detection Model 62](#_Toc528077779)

[4.4.4 Test your re-trained Image Object Detection Model 64](#_Toc528077780)

[5 Bring Your Own Model (BYOM) 66](#_Toc528077781)

[5.1 Generate Access Token 67](#_Toc528077782)

[5.2 Upload and Deploy your Model 68](#_Toc528077783)

[5.3 Test your Model 73](#_Toc528077784)

[6 Appendix 76](#_Toc528077785)

[6.1 Generate Access Token via Postman 76](#_Toc528077786)

[6.2 Setup ML Foundation in an SAP Cloud Platform Trial Account 77](#_Toc528077787)

[6.3 Re-train Image Classification with sapml 82](#_Toc528077788)

# Authorization

This chapter is not an exercise. It is just for your information with which user you will work during the whole exercise document.

We created a user for you that you will use to logon to

* SAP API Business Hub ([**http://api.sap.com**](http://api.sap.com))
* SAP Cloud Platform (SCP) Cockpit ([**https://cloudplatform.sap.com**](https://cloudplatform.sap.com))
* SCP – Trial Accounts **(**[**https://account.hanatrial.ondemand.com**](https://account.hana.ondemand.com)**)**
* SCP – Productive Global Accounts **(**[**https://account.hana.ondemand.com**](https://account.hana.ondemand.com)**)**
* SAP Cloud Foundry via Cloud Foundry Command Line Client (CF CLI)

This user should have all the necessary privileges to perform all parts of the following exercises. The username you should use is associated with your group and follows this pattern:

ml-train+us-XXX@sap.com

where xx needs to be replaced by your group number, e.g. if your group number is 001, then you should take the user **ml-train+us-XXX@sap.com**

Please make sure that you work with the correct username to avoid conflicts with other participants. Take note of the following credentials, you will be using them during all exercises.

**Username:** ml-train+us-XXX@sap.com

**Password:** <specific to your username> 🡪 provided by the trainer

# Try the SAP Leonardo Demo Page (optional)

**Objective**

In this exercise you can experience how easy it is to use the available SAP Leonardo Machine Learning foundation functional services*.* The models are already pre-trained and can be tried out on the demo web page.

## Try out the Person Finder

Estimated time: 10 minutes

| Explanation | Screenshot |
| --- | --- |
| 1. Open your browser and type in:   <https://leo-mlp-demo-solutionexper-v3.cfapps.eu10.hana.ondemand.com>  This will open the SAP Leonardo Machine Learning Foundation Demo page.  Click on tile *Face Detection* |  |
| 1. On the upcoming screen click *Live Demo* |  |
| 1. Scroll down and click *Find Person* |  |
| 1. Click the image above *Step 1* to select an image |  |
| 1. In the “Browse File” dialogue select image bill.jpg from the folder *Exercise0*. |  |
| 1. Click the middle box and select image people.jpg out of the same folder.   Click *Find out now* |  |
| **Information step:** The face detection service will first find the faces in the images and then extract the features of the faces in feature vectors. The face feature vector of the first image is then compared with the face feature vectors of the second image, using the similarity scoring service. The face with the best similarity is selected and the similarity is shown as confidence level of the match on the right-hand side. |  |

## Try out the further demo scenarios

Estimated time: 10 minutes

| Explanation | Screenshot |
| --- | --- |
| 1. Open your browser and type in:   <https://leo-mlp-demo-solutionexper-v3.cfapps.eu10.hana.ondemand.com>  This will open the SAP Leonardo Machine Learning Foundation Demo page.  Click on any of the tiles.  On the upcoming screen click *Live Demo* |  |
| 1. Depending on the demo you typically need input data like text snippets or images. We’ve prepared sample files that can be found in the folder *Exercise0*.   Upload the respective files and play around with some of the services. |  |

Summary

You have completed the exercise!

You are now able to:

* Use the SAP Leonardo Demo page that illustrates various machine learning scenarios.
* Get an idea how various services can be combined to solve a real business problem.

# Ready-to-use Services via API Business Hub

## Use SAP Leonardo ML Document Feature Extraction

Estimated time: 15 minutes   
  
**Objective**

In this exercise you can experience how easy it is to use the available SAP Leonardo Machine Learning foundation services on SAP API Business Hub*.* The models are already pre-trained and can be tried out in a sandbox system. Here we use the *Document Feature Extraction API*.

**Example**: Inference Service for Document Feature Extraction

**SAP API Business Hub:** <https://api.sap.com>

Required resources for this exercise:

DFE\_articles.zip – zip file with .txt files about document feature extraction.

| Explanation | Screenshot |
| --- | --- |
| 1. Open SAP API Business Hub in your browser and *search for*Leonardo and *press Enter*.   <https://api.sap.com> |  |
| 1. Choose SAP Leonardo Machine Learning Foundation – Functional Services. |  |
| 1. Choose *Log On*.   **Note:** Currently there are 23 functional services listed. |  |
| 1. Enter your user-id and password (here: [ml-train+us-XXX@sap.com](mailto:ml-train+us-XXX@sap.com)) and logon.   **Note:** XXX is your 3-digit user which is assigned during the workshop.  Accept the Terms & Conditions and choose to remember the login data. | ml-train+us-XXX@sap.com |
| 1. Search for Document Feature Extraction, press Enter, and choose the Inference Service for Document Feature Extraction. |  |
| 1. Under API References, open the *POST* service to test an inference sync/call.   And press the *Try out* button to use the API. |  |
| 1. Under *files* press the Button *Choose File* to open the File Explorer. Choose the prepared DFE\_articles.zipfile containing various texts from newspaper and recipes from the folder *Exercise1*.   Start the API call by pressing the *Execute* button. |  |
| 1. Validate the result in the Response Body.   The response contains a vector for each file in the input ZIP-file. Each vector consists of 200 numbers. |  |
| **Information Step:**  We will do Similarity Scoring based on this result. But you do not have to save the result in the correct format in .txt files, because we prepared these files already for you for the next exercise. |  |

Explanation

The document feature extraction API returns a “feature vector”. Every component represents a certain feature, which was identified by the underlying machine learning model. Even for a data scientist with access to the model it is very hard to understand what every component/number of that vector represents. This knowledge is not required here as we will see in the next exercise how to leverage those vectors to come to useful conclusions.

Summary

You have completed the exercise!

You are now able to:

* Browse through API Business Hub to find the latest functional and business services of SAP Leonardo ML foundation.
* Test SAP Leonardo ML foundation services directly on API Business Hub.
* Transform unstructured text data into a fixed length vector using document feature extraction.

## Use SAP Leonardo ML Similarity Scoring

Estimated time: 15 minutes   
  
**Objective**

In this exercise you will use the *Similarity Scoring API* on SAP API Business Hub, where it can be directly tested. The Similarity Scoring API compares vectors using cosine distance to calculate similarity score of one or more input vectors.

**Example**: Inference Service for Similarity Scoring

**SAP API Business Hub:** <https://api.sap.com>

Required resources for this exercise:   
SimScore\_Vector.zip – zip file with .txt files about Similarity Scoring.

DFE\_vectors.zip – zip file with .txt files containing the output of document feature extraction.

| Explanation | Screenshot |
| --- | --- |
| 1. Open SAP API Business Hub in your browser and choose *Log On*.   <https://api.sap.com>  **Note:** In case you are already logged in, you can skip steps 1 and 2 and continue with step 3. |  |
| 1. Enter your user-id and password (here: [ml-train+us-XXX@sap.com](mailto:ml-train+us-XXX@sap.com)) and logon.   **Note:** XXX is your 3-digit user which is assigned during the workshop.  If a pop-up windows occurs, accept the Terms & Conditions and choose to remember the login data. | ml-train+us-XXX@sap.com |
| 1. Search for “Similarity Scoring” and press *Enter*. |  |
| 1. Choose the *Inference Service for Similarity Scoring*. |  |
| 1. Under API References, open the *POST* service to test an inference sync/call.   And press the *Try out* button to use the API. |  |
| 1. Under *files* press the Button *Choose File* to open the File Explorer. Choose the prepared SimScore\_Vector.zip file containing simple vectors from the folder *Exercise1*. |  |
| 1. Under the “parameters” section enter {"numSimilarVectors":2} for the “options” parameter.   **Note:** You can copy it from the Example on the web page.  Start the API call by pressing the *Execute* button. |  |
| 1. Validate the result in the Response Body. For each vector it contains the similarity scoring with the other two vectors. |  |
| 1. Take a look at the vectors in the ZIP file “SimScore\_Vector.zip”. Does the result of the Similarity Scoring make sense?  Hint: cos(0°) = 1 cos(45°) = 0.707… cos(90°)= 0 |  |
| 1. Repeat the exercise with the DFE\_vectors.zip ZIP-file. Make sure to adjust the “options”-parameter numSimilarVectors  Compare your results with the matrix nearby. Can you find the similarity score between the newspaper article and the song? |  |

Explanation

Similarity Scoring of two vectors returns the cosine of the angle between them. When interpreting the result, keep in mind that a cosine close to 1 means that the vectors are almost identical. While a value close to 0 means that the vectors are very different.

Vectors are the most important artifacts in machine learning. Every kind of input such as text, image or table record will be represented as a vector of features. All calculations will be done on those vectors. By comparing two vectors we automatically compare two corresponding artifacts (e.g. two images or documents) and can say how similar is the content of these documents or the object in these images.

Summary

You have completed the exercise!

You are now able to:

* Browse through API Business Hub to find the latest functional and business services of SAP Leonardo ML foundation.
* Test SAP Leonardo ML foundation services directly on API Business Hub.
* Compute angles between (feature) vectors using Similarity Scoring.

## Use SAP Leonardo ML Topic Detection

Estimated time: 10 minutes   
  
**Objective**

In this exercise you can try out the *Topic Detection API*, which is able to find topics and keywords in documents without being trained by customers.

**Example**: Inference Service for Topic Detection

**SAP API Business Hub:** <https://api.sap.com>

Required resources for this exercise:

Topic\_Detection.zip – zip file with .txt files about topics.  
DFE\_articles.zip – zip file with .txt files about document feature extraction.

| Explanation | Screenshot |
| --- | --- |
| 1. Open SAP API Business Hub in your browser and choose *Log On*.   <https://api.sap.com>  **Note:** In case you are already logged in, you can skip steps 1 and 2 and continue with step 3. |  |
| 1. Enter your user-id and password (here: [ml-train+us-XXX@sap.com](mailto:ml-train+us-XXX@sap.com)) and logon.   **Note:** XXX is your 3-digit user which is assigned during the workshop.  If a pop-up windows occurs, accept the Terms & Conditions and choose to remember the login data. | ml-train+us-XXX@sap.com |
| 1. Search for “Topic Detection” and press *Enter*. |  |
| 1. Choose the *Inference Service for Topic Detection*. |  |
| 1. Under API References, open the *POST* service to test an inference call.   And press the *Try out* button to use the API. |  |
| 1. Under *files* press the Button *Choose File* to open the File Explorer. Choose the prepared DFE\_articles.zip file containing text files about computer science and pies from the folder *Exercise1*. |  |
| 1. Under *options*, copy and adjust the parameters from the Example and paste it into the Value field:  {"numTopics":3, "numTopicsPerDoc":2, "numKeywordsPerTopic":15}   Start the API call by pressing the *Execute* button. |  |
| 1. Validate the result in the Response Body.   The response contains keywords for all submitted files in the zip file.  Test it again with changed options. Therefore change the values for numTopics to 2 and numTopicsPerDocs to 1.  Try it out!  What has changed? |  |

**Explanation:** The service **Topic Detection API** finds ***numKeywordsPerTopic*** keywords and ***numTopics*** across all documents. The topics get a number, starting with 0. So, if you define 3 Topics, they will be numbered with 0, 1, 2. With ***numTopicsPerDoc*** you define how many of the topics of the entire document corpus can be found in one document. The algorithm chooses this number of topics which fits best and assigns them a score which is not normalized to 1.

Summary

You have completed the exercise! You are now able to:

* Browse through API Business Hub to find the latest functional and business services of SAP Leonardo ML foundation.
* Test SAP Leonardo ML foundation services directly on API Business Hub.
* Understand the Topic Detection Model.

## Use SAP Leonardo ML Image Processing APIs

Estimated time: 15 minutes   
  
**Objective**

In this exercise you can experience how easy it is to use the available SAP Leonardo Machine Learning foundation services on SAP API Business Hub*.* The models are already pre-trained and can be tried out on the web page.   
In this exercise, you have time to test some of the remaining APIs on the Business Hub. In the following instructions are some suggestions.

**Example**: Image Processing APIs

**SAP API Business Hub:** <https://api.sap.com>

### Use SAP Leonardo ML Image Classifier Service

| Explanation | Screenshot |
| --- | --- |
| 1. Use the “Inference Service for Customizable Image Classification” to detect objects in pictures.  Browse the internet for some images and try them out!  The service works very well with different dog breeds. |  |

### Use SAP Leonardo ML Image Feature Extraction API

| Explanation | Screenshot |
| --- | --- |
| 1. Use the “Inference Service For Customizable Image Feature Extraction” to compute the feature vector of an image.  Again, as in Exercise 1.1, the vector itself does not carry any meaning to us, but can be used for comparisons. |  |

### Use SAP Leonardo ML Optical Character Recognition

| Explanation | Screenshot |
| --- | --- |
| 1. Use the “Inference Service for Optical Character Recognition (OCR)” to extract texts from images.  Browse the internet for some images or take a picture with your camera to try it out.  You can also use the one that is provided in the corresponding exercise folder. |  |

# Ready-to-use Services in application development

## Setup Development Environment in SAP Cloud Platform

For the exercises where we develop SAPUI5 applications, we will work with the SAP Web IDE in SAP Cloud Platform. To save time, we prepared a project for you so that we can focus on the ML Foundation parts. This project needs to be imported into your SAP Web IDE workspace. You will extend the applications included in the project to test the SAP Leonardo ML services.

In this chapter, you will initialize the SAP Web IDE and import the given project.

Required resources for this step:

MLFSAPUI5\_Project\_Exercises.zip – prepared SAPUI5 app for exercise

MLFSAPUI5\_Project\_Solution.zip – solution of this exercise

| Explanation | Screenshot |
| --- | --- |
| 1. Open **Firefox** as Chrome will use SSO for SAP Employees. **Customers can use Chrome.**   Login to SAP Cloud Platform Cockpit via [http://cloudplatform.sap.com](http://cloudplatform.sap.com/).  Login with your given e-Mail  **ml-train+us-XXX@sap.com** and the password provided by the trainer. |  |
| 1. Navigate to Regions >  Trial > *Europe (Rot) – Trial*.   Note:  We will use the SAP Web IDE version provided via the SCP Neo environment. |  |
| 1. You will find a link to your trial account. Click on this trial account. |  |
| 1. In the left menu bar – navigate to Services and search für  *Web IDE*.   You will get to flavours of the SAP Web IDE. We will use the  *SAP Web IDE.* |  |
| 1. Click on   which will open up the SAP Web IDE. |  |
| 1. Navigate to the Development perspective, it is the second icon in the left toolbar.   This will open your empty Workspace.  Now we will import the prepared project.  **Note**: In case your workspace is not empty. Please right-click all projects and delete them. |  |
| 1. Right-click on Workspace and select Import > From file System. |  |
| 1. Browse in your file system and select MLFSAPUI5\_Project\_Exercise.zip to import the project.   We will need this project later in this document when you develop SAPUI5 apps to call ML Foundation services. |  |
| 1. You should see a files and folder structure similar to the one shown on this screenshot. |  |

## Use SAP Leonardo ML Image Classification with SAPUI5

Overview

Estimated time: 25 minutes   
  
**Objective**

In this exercise you will learn how to quickly integrate the **Image Classification** SAP Leonardo Machine Learning Functional Services published from the SAP API Business Hub sandbox in a SAPUI5 application.

**Example**: Image Classification API

**SAP API Business Hub:** <https://api.sap.com>

Prerequisite:

You completed the preparation and imported the project ***MLFSAPUI5\_Project\_Exercise.zip*** into your SAP Web IDE workspace (see chapter 2.1).

**Exercise Description** - SAP Leonardo ML Foundation service consumption

* *Import a SAPUI5 application calling a REST service (🡪 see prerequisite)*
* *Create a destination in SCP Cockpit to your ML Service endpoint in API Business Hub*
* *Use the API Key for the image classifier service from SAP API Business Hub*
* *Adjust your prepared SAPUI5 application*
* *Run the SAPUI5 application and test your ML Services.*

Required resources for this exercise:

* MLFSAPUI5\_Project\_Exercise.zip (already imported in SAP Web IDE before)
* sapui5\_neo-app\_snippet.json
* keyboard.jpg, laptop.png, …
* Optional: prepared destination “sapui5ml-api”

**Create Destinations for ML Services in API Business Hub**

| Explanation | Screenshot |
| --- | --- |
| 1. Open a new tab in your Firefox web browser and go to  [**SAP Cloud Platform Cockpit**](http://account.hanatrial.ondemand.com/cockpit)   <https://cloudplatform.sap.com> > Login in case you are not already logged in.  Or use direct link: <http://account.hanatrial.ondemand.com/cockpit>  Logon using the given credentials and **not** **SSO** [only for SAP employees]. | <username> |
| 1. Go to Regions > Trial >  **Europe (Rot) – Trial.** |  |
| 1. You should see this screen. Click on the trial account. |  |
| 1. On the left side bar, you can navigate in Connectivity > Destinations |  |
| 1. On the Destinations overview page, click on *New Destination.*   **Note**: You can also import the prepared destination with the name “**sapui5ml-api**”. |  |
| 1. Enter the following information:   **Name**: sapui5ml-api  **Type**: HTTP  **Description**: SAP Leonardo Machine Learning APIs  **URL**: <https://sandbox.api.sap.com/ml>  **Proxy Type**: Internet  **Authentication**: NoAuthentication |  |
| 1. Click on the “New Property” Button and add the following properties to the destination:   **WebIDEEnabled = true**  Click on Save. |  |
| 1. Then You can use the **Check Connection** button to validate that the URL can be accessed. Or choose the icon highlighted on the right.   The response should be: Connection to "sapui5ml-api" established. Response returned: "404: Not Found" |  |

**Adjust the prepared SAPUI5 application to access your ML foundation services**

| Explanation | Screenshot |
| --- | --- |
| 1. Open your SAP Web IDE. Navigate to your project that you imported before. |  |
| 1. Open the neo-app.json file located under **Workspace > MLFSAPUI5\_Project\_Exercise > neo-app.json** |  |
| 1. Edit the neo-app.json file and add a new route to the new destination with the name “**sapui5ml-api**” we created in the cloud platform cockpit before.   Therefore, please use the code snippet in file  *sapui5\_neo-app\_snippet.json*.  It should look like the one shown on the right side. |  |
| 1. Scroll to the end of the neo-app.json and add the String "APIKey" to the headerWhiteList.   Then click on the Save Button button (or press CTRL+S). |  |
| 1. Open the settings.json file located under **Workspace > MLFSAPUI5\_Project\_Exercise > webapp > model > settings.json** |  |
| Before we proceed with editing the settings.json, we need to get the API Key for the ML services from the API Business Hub, explained in the next steps. |  |
| 1. Go to <https://api.sap.com> and search for *Machine Learning Foundation Functional Services*. |  |
| 1. Then just click on SAP Leonardo Machine Learning Foundation - Functional Services. |  |
|
| 1. Scroll down and click on the  **Inference Service for Customizable Image Classification**.   Login in case you are not already logged in. |  |
| 1. When using any of the APIs outside of the SAP API Business Hub, an application key will be needed in every request header of your API calls.   To get your API key click on the „*Show API Key*“ button on top of the page.  Here you see your key. Choose the  *“Copy Key and Close*” button to copy it to your clipboard. |  |
| Now – go back to your SAP Web IDE in SAP Cloud Platform, where the settings.json is still open. |  |
| 1. Edit the settings.json.   For the url “Image Classification” replace the  **<<<< COPY YOUR API KEY >>>>** with the API key we retrieved in the step before in the API Business Hub. Then click on the Save Button button (or press CTRL+S). |  |
| 1. Click on the **Run** icon Run Applications or press ALT+F5.   Either drag & drop the selected image onto the icon or press the icon to open a file select dialog.  Choose *tennis\_ball.jpg*, *keyboard.jpg,* *laptop.png* and *iPhoneX* as test images and analyze the results. |  |
| 1. The micro charts the 5 most probable classifications for the image. To see the actual response from the server you can press the “View JSON” Button. |  |

Summary

You have completed the exercise!

You are now able to:

* Use the SAP Web IDE to import a SAPUI5 application
* Create a destination that points to your ML service API endpoint in SAP API Business Hub
* Adjust a prepared project using a configured destination calling REST services from ML foundation

# SAP ML Foundation in SAP Cloud Foundry

## Install Cloud Foundry Command Line Interface (CF CLI)

To work with the Cloud Foundry Command Line Interface (CF CLI) you first need to install it:

* Go to <https://help.sap.com/viewer/65de2977205c403bbc107264b8eccf4b/Cloud/en-US/856119883b8c4c97b6a766cc6a09b48c.html>   
  and choose ***Download and Install the Cloud Foundry Command Line Interface***
* Read the procedure for ***Windows*** respectively ***Mac OS*** and do the installation
  + For Microsoft Windows, download the Windows 64-bit installer and start it
  + For Mac OS, download the PKG file and start the installer

## Create ML Foundation Service Instance & Service Key

You used the Cloud Foundry user to access the SAP API Business Hub and the SAP Cloud Platform – Neo environment.

Now you will learn how to work with ML Foundation Services that are available within your SAP Cloud Platform Global Account.

Please make sure that you work with the correct username to avoid conflicts with other participants. Take note of the following Cloud foundry credential, you will be using it in the following steps.

Organisation and space will be automated configured/available during the login process as there is just one space and org for each user.

**Username:** ml-train+us-XXX@sap.com

**Password:** <specific to your username> 🡪 provided by the trainer

Just for your information:

**API endpoint**: api.cf.us10.hana.ondemand.com

**Subaccount**: ml-train-XXX

**Space:** training

**SAP Cloud Foundry Org and Space via SCP Cockpit**

| Explanation | Screenshot |
| --- | --- |
| 1. Open a new browser window and enter the following URL: [https://account.hana.ondemand.com](https://account.hana.ondemand.com/)   Choose the *Log on* button. |  |
| 1. Login with your given e-Mail  ml-train+us-XXX@sap.com and the password provided by the trainer. | <username> |
| 1. Confirm the EU Access Mode. |  |
| 1. Click on your Global Account 🡪 **SAP\_ML\_Demo** |  |
| 1. Click on the Subaccount – which is your Organisation (Cloud Foundry org) 🡪 **ml-train-XXX**   Where xx is your group number. |  |
| 1. Here you see your Cloud Foundry details like name of your Organization, number of Spaces (should be 1 at the beginning) and the API endpoint. | api.cf.us10.hana.ondemand.com |
| 1. Click on **Spaces** in the left menu bar and then on your Space with the name **training**. |  |

**Create Service Instance and Service Key for ML Foundation**

| Explanation | Screenshot |
| --- | --- |
| 1. Within your space navigate to Services > Service Marketplace |  |
| 1. Here you should find the service ml-foundation as a tile.   Click on that link **ml-foundation**. |  |
| 1. On the left menu bar click on *Instances*.   You will find a button  . Click on it. |  |
| 1. Select the Plan “standard” and choose Next. |  |
| 1. Next  (no parameters to specify) |  |
| 1. Next |  |
| 1. Give your instance a new name, like ml\_instance\_xxx where xx is your number.  Click Finish. |  |
| 1. Click on the newly created instance. |  |
| 1. Select Service Keys in the left menu bar and choose the button *Create Service Key* to create a new service key for your instance. |  |
| 1. Give it a name like ml\_service\_xxx where xx is your number.   Click Save to save your newly created service key. |  |
| 1. You get a similar screen like the one shown on the right side.   Note:  A service key enables our ML Foundation Service to be used outside the CF environment. In this exercise, we are creating a key to be used by a program or e.g. Swagger UI or Postman.  The service key contains all the URLs and credentials (clientid and clientsecret) required for you to access the ML Foundation Services running for your productive account.  Please keep this browser window open. You will need it in the next section. | **Note**: Copy the complete service key and paste it in a text file. You will need theses values and URLs serveral times during the exercises. |

### Create Service Instance and Service Key for ML Foundation Beta (optional)

Note: this chapter is only required if you want to use beta services of ML Foundation, like as required for Exercise 4.4 Customizable Image Object Detection.

| Explanation | Screenshot |
| --- | --- |
| 1. Within your space navigate to Services > Service Marketplace |  |
| 1. Here you should find the service ml-foundation-beta as a tile.   Click on that link **ml-foundation-beta**. |  |
| 1. Repeat all steps described above for creating a service instance and a service key.   Please use the following naming:  instance:  mlbeta\_instance\_xxx  service key: mlbeta\_servicekey\_xxx  In the service key you will find the beta services of ML Foundation for testing puposes – not for productive usage! |  |

Now your environment is prepared for productive usage of ML Foundation in SAP Cloud Foundry and you can start the exercises.  
  
**Have fun!**

# Customizable Services

## Generate Access Token

For exercises 4 and 5 we will need a OAuth2 token to access the ML Foundation services in our productive Cloud Foundry environment. We prepared a small helper application. Please use the following link to access this token generation app. You need to be logged in to your SAP Cloud Foundry account, to access it:

<https://generate_ml_token.cfapps.eu10.hana.ondemand.com/>

**Note**: The Generate Token is a tool that we provide during this workshop. It is not part of our standard ML Foundation service offering. In the appendix, you will find a description on how to generate the token by yourself with a tool like Postman.

| Explanation | Screenshot |
| --- | --- |
| 1. A new window comes up where you need to enter the information provided via your service key:   Authentication URL Client-ID Client-Secret  See next step. |  |
| 1. Please go back to your service key and copy and paste the three required values into the Token Generator app. |  |
| 1. Click on |  |
| 1. A token with the Prefix “Bearer” is generated. Use the ***Copy To Clipboard!*** button with which you can copy and paste it.   **Note**: You do not need to copy the token now. But we will reference this section when you need the token to access the ML foundation services. | C:\Users\d037045\AppData\Local\Temp\SNAGHTML1ad14b.PNG |

## Customizable Image Classification

Overview

Estimated time: 45 minutes

Objective

In this exercise you learn how to 'fine-tune' a generic machine learning model to a custom use case. You can use this service without being a Machine Learning Expert. The base model is provided by SAP. The Retrain Service adapts the base model to a specific model using training data provided by the customer.

As a result, the classification results for the customer data-set are significantly improved compared to the generic base model.

**Exercise Description** - SAP Leonardo ML Foundation Retrain Service

* *Login to SAP CF using the CF CLI*
* *Deploy the Retraining App into your SAP CF account*
* *Create Retrain Service Instance and Service Key in SCP Cockpit (already done)*
* *Upload the training data. (Note: the training data is already uploaded for this exercise)*
* *Perform retraining job using the test retraining app*
* *Deploy the retrained model using the test retraining app*
* *Test the retrained model using the test retraining app*
* *Use the deployed retrained model for inference (with the prepared SAPUI5 app)*

**Prerequisite**

You need to create a service instance and service key for the ML foundation service in your SAP Cloud Foundry space. Done in previous exercise.

For this exercise, you will need the retraining app that you will deploy into your space. This is an application for training and test purpose. Note: it is not part of the standard MLF product delivery.

The data for retraining is already uploaded to the file system associated with your Cloud foundry ML Foundation service account. This is a step that you need to perform as well in a productive environment or POC. We skipped it during this workshop as it would take some time.

As the BYOM service is running in the productive Cloud Foundry environment, you will need an Access Token to run the service. Please use the Generate Token helper tool in order to get the OAuth2 token.

**Access to SAP Cloud Foundry Space:**

**Username:** ml-train+us-XXX@sap.com

**Password:** <specific to your username>

**API endpoint**: api.cf.us10.hana.ondemand.com

**Subaccount**: ml-train-XXX

**Space:** training

Cloud Foundry Login command:

cf login -a api.cf.us10.hana.ondemand.com -u ml-train+us-XXX@sap.com

Required resources for this exercise:

* retraining-app-master.zip
* Sample images: BMW.jpg, SAP\_perturbed.jpg, SAP\_pertubed\_5.jpg, SAP\_WDF19.jpg, …

### Deploy Re-training Application

For this exercise we provide an application to work with the SAP MLF re-training service for image classification. This application is a test UI and not delivered as standard product.

| Explanation | Screenshot |
| --- | --- |
| 1. Download the zip file  “retraining-app-master.zip” and unpack it in your file system. |  |
| 1. Open the manifest.yml with a text editor (e.g. Wordpad).   Replace the following fields:  **Name**: “retrainingapp\_xx” **services**: ml\_instance\_xxx (where xx is your group number) **SERVICE\_BROKER\_NAME**: ml\_instance\_xx (where xx is your group number) |  |
| 1. Open a command prompt (in the windows start menu enter “cmd” and hit Enter) |  |
| 1. Navigate to the folder where the manifest.yml is located.   Login to Cloud Foundry by typing the following command:  cf login -a api.cf.us10.hana.ondemand.com -u [ml-train+us-XXX@sap.com](mailto:ml-train+us-XXX@sap.com)  (where xxx is your group number)  When prompted, enter your password. Note that characters are not shown on the screen! You can paste by doing a right click with the mouse.  Now you can deploy the application using the command:   cf push |  |
| 1. The application will be created in your space with a route that you can access via URL later. | XXX  XXX  XXX  XXX  XXX  XXX |
| 1. Wait until this process has ended. You should see a similar screen where you find the state “running” and the URL of your application. | XXX |
| 1. Copy this URL and prefix it with “https://” and copy it into a new browser window.   (You can copy in Windows Command Prompt by selecting the characters with your mouse, then confirm with pressing *ENTER*)  The Retraining App should come up.  You have successfully deployed your first UI application on SAP Cloud Platform! | XXX |

### Training data

For this re-training exercise the required data was uploaded before, to save some time. During this exercise you can have a look at the data structure and content of the training images.

| Explanation | Screenshot |
| --- | --- |
| 1. Navigate to “Manage Training Data”. |  |
| 1. Open the Brands folder |  |
| 1. Here you see the folder structure with test, training and validation sub folders. |  |
| 1. Within each folder you find the 5 categories: Adidas, Apple, BMW, Coca-Cola and SAP. In each of these folders the according images are stored. |  |

### Re-train the Image Classifier Model

In the next section, you will retrain the pre-trained Image Classification Model that is behind the exposed corresponding service. The data for retraining is already uploaded to your filesystem on the cloud data center. You will start a retraining job based on that data. Your re-trained model is created, automatically uploaded to the model repository.

| Explanation | Screenshot |
| --- | --- |
| 1. Navigate back to the entry page and select “*Manage Retraining Jobs*”. |  |
| 1. Create a new retraining job by clicking the + sign. |  |
| 1. Select the Brands/ folder and give it a model name (e.g. brands-xxx, where xxx is your group number).   Click the *Create* button to start the re-training job.  With that the job is scheduled for training in the background. The ML Foundation infrastructure will automatically assign a free virtual machine. | brands-xxx |
| 1. As soon as a machine was assigned, the re-training job is started and has the status “Pending” and then “Running”   Click the icon to enable auto-refresh of the screen.  During the “Running” phase, the generic base model is being re-trained with the photos provided in the Brands dataset.  **Note:** Overall the whole process can take between 5 to 20 minutes. If you want to bridge the waiting time you may already start with Exercise 2 |  |
| 1. In the overview page you can see, that there is one job in status “Running”. Wait until it has the status “Succeeded”. |  |
| 1. When you see, that your job is now listed with the status “Succeeded”, navigate to its details by clicking the arrow at the end of the row. |  |
| 1. Here you see some details of your re-training job. Start date, end date, number of epochs, name and version of your new model. |  |

### Deploy the re-trained Image Classifier Model

In the next section, you will deploy the model that was created and uploaded after successful training to the model repository. Deploy a model means, a serving container is created that can be used for inference.

| Explanation | Screenshot |
| --- | --- |
| 1. Navigate to “Manage Retrained Models”. |  |
| 1. Select the name of your model (*brands-xxx*) in the list of undeployed models. Click the *Deploy* button. |  |
| 1. It will move to the status “Pending”. Please wait a bit until your model is successfully deployed. Click refresh from time to time. |  |
| 1. As result, you should see your model in the section “Succeeded”. |  |
| 1. When clicking on the arrow at the end of the row of your model, you come to a testing screen. Here you can do the inference with images compared to the standard model. |  |
| 1. For quick testing purposes, the retraining application allows to directly test your model.   To do so click *Select Image* and select one of the images provided in folder *Exercise 4*.  Click the checkbox *Compare with Generic Model*  Then click the *Inference* button |  |
| 1. Compare the results of your model (left-hand side) and the generic base model (right-hand side).   Did your model achieve higher accuracy when uploading brand logos? |  |

Summary

You have completed the exercise!

You are now able to:

* Use the re-trainable image classification service to
* Retrain the model based on your custom data
* Deploy your retrained Model
* Use the retrained Model for inference and see improvements compared to the base model

## Customizable Text Classification

Overview

Estimated time: 45 minutes

We'll see how you can customize the text classification service to return understand positive and negative sentiments on automotive reviews.

The steps we follow are highlighted below.

Objective

In this exercise you learn how to 'fine-tune' a generic machine learning model to a custom use case. You can use this service without being a Machine Learning Expert. The base model is provided by SAP. The Retrain Service adapts the base model to a specific model using training data provided by the customer.

Thus, the classification results for the customer data-set are greatly improved compared to the generic base model.

**Exercise Description** - SAP Leonardo ML Foundation Retrain Service

* *Login to SAP CF using the CF CLI*
* *Deploy the Retraining App into your SAP CF account*
* *Create Retrain Service Instance and Service Key in SCP Cockpit (already done)*
* *Upload the training data. (Note: the training data is already prepared)*
* *Perform retraining job using the swagger UI*
* *Deploy the retrained model using the swagger UI*
* *Use the deployed retrained model for inference (with a prepared SAPUI5 app)*

**Prerequisite**

You need to create a service instance and service key for the ML foundation service in your SAP Cloud Foundry space. Done in previous exercise.

The data for retraining is already prepared. You need to upload it to the file system associated with your Cloud foundry ML Foundation service account.

As the re-training service is running in the productive Cloud Foundry environment, you will need an Access Token to run the service. Please use the Generate Token helper tool in order to get the OAuth2 token.

**Access to SAP Cloud Foundry Space:**

**Username:** ml-train+us-XXX@sap.com

**Password:** <specific to your username>

**API endpoint**: api.cf.us10.hana.ondemand.com

**Subaccount**: ml-train-XXX

**Space:** training

Cloud Foundry Login command:

cf login -a api.cf.us10.hana.ondemand.com -u ml-train+us-XXX@sap.com

Required resources for this exercise:

* template.zip

### Re-train the Text Classifier Model

After you uploaded the training data, you can now retrain the pre-trained Text Classification Model that is behind the exposed corresponding service. You will start a retraining job based on that data. Your re-trained model is created, automatically uploaded to the model repository.

| Explanation | Screenshot |
| --- | --- |
| 1. Navigate in the SCP Cloud Foundry Cockpit to your Service Key.   For text classification retraining you’ll use TEXT\_LINEAR\_RETRAIN\_API\_URL in the service key.  Mark this URL and copy it. | xx10 |
| 1. Open a new browser window, paste and open the copied URL. A Swagger UI is opened up  The Swagger UIs allows the development team to visualize and interact with APIs, without having implementation logic in place.   It provides an overview and documentation of all API operations, as well as sample requests and responses.  (**Note**: it might change in the future). |  |
| 1. Open the first entry > *jobs* and select the ***POST*** method.   Click on *Try it out*. |  |
| 1. Now you can submit the retraining job from swagger UI /jobs specifying the folder name under which the training data was uploaded (here: Sentiment-Twitter) and the model name (sentiment\_xx where xx is your group number).   {  "dataset": "**Sentiment-Twitter**",  "modelName": "**sentiment\_xx**",  "preprocessingLanguage": "en",  "completionTime": 24,  "memory": 8192  }  (Replace the XXX with your)  To generate the bearer token, you can use the helper app:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/>  Click on the *Execute* button. |  |
| 1. Note the job id returned in the server response. We will need it in the next step.   The job is automatically scheduled and processed in background and is now starting to re-train the base model with the custom provided data of folder “Twitter-Sentiment” |  |
| 1. You can query the job ID to check the status of training job.   Open the second GET method under /jobs where you can get the status of a retraining job with a given ID.  Click the *Try it out* button. |  |
| 1. Paste the job id that you copied before. Enter your bearer token and click on the *Execute* button.   You may click the *Execute* button multiple times to retrieve the latest job status.  Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/> |  |
| 1. It might take some time for the training container to get allocated and for the training job to run.     Logs of the training job are visible under data\jobs\<id> on your allocated file system storage (you can use sapml CLI).  After a while (approx. 10 minutes) you should see your job with the status “SUCCEEDED”. |  |

### Deploy the re-trained Text Classifier Model

In the next section, you will deploy the model that was created and uploaded after successful training to the model repository. Deploy a model means, a serving container is created that can be used for inference.

| Explanation | Screenshot |
| --- | --- |
| 1. Once the job has succeeded you deploy the model using deployment api. You are still in the swagger UI for text retraining and navigate to deployments. Here you can open the POST method to deploy a retrained model. Click on *Try it out*.   Specifying modelname and model version. (model name is the model name used for retraining job) |  |
| 1. Specify the modelName. This is the one that you defined for the re-training you.  Here: sentiment\_xx where xx is your group number..   Version should be 1 as you run this re-training job for the first time.  Enter the bearer token and click on the *Execute* button.  Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/> |  |
| 1. Have a look at the server response. The deployment job gets an ID. Please copy it – we will need it in the next step. |  |
| 1. Query the deployment status using GET /deployments/{id}.   Enter your job id and the bearer token and click on the *Execute* button.  Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/> |  |
| 1. Have a look at the server response. The state should change to SUCCEEDED once model is deployed successfully. |  |
| 1. Once you have the model successfully deployed it can be used for inference by invoking TEXT\_CLASSIFIER\_URL from the service key.   Copy the URL into a new browser window. | xx10 |
| 1. Here you enter the model name, model version, bearer token and the text for which you want to test the classification. E*xecute*.   Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/>  Note: Currently (29.06.2018) not working. Will be fixed soon. |  |

### Deploy test app & test re-trained Text Classifier Model

| Explanation | Screenshot |
| --- | --- |
| 1. Download the zip file  “template.zip” and unpack it in your file system. |  |
| 1. Open the *manifest.yml* (right-click 🡪 *Edit with Notepad++)* |  |
| 1. In the manifest.yml   Replace the following fields:  name: retrain\_text\_app\_xx host:host\_xx services: ml\_instance\_xx MODEL\_NAME: sentiment\_xx   (where xx is your group number) |  |
| 1. Open a command prompt and navigate to the folder where the manifest.yml is located.   Login to Cloud Foundry:  cf login -a api.cf.us10.hana.ondemand.com -u ml-train+us-XXX@sap.com  (where xxx is your group number)  When prompted, enter your password. Note that characters are not shown on the screen! |  |
| 1. Enter your password. Now you can deploy the application using the command:   cf push |  |
| 1. Wait until this process has ended. You should see a similar screen where you find the state “running” and the url of your application. | **us10.hana.ondemand.com** |
| 1. Copy this URL and prefix it with “https://” and copy it into a new browser window.   (To copy from Windows Command Prompt select the URL with your mouse and press ENTER once)  The Sentimental Classification test App will come up. |  |
| 1. In case you want to launch the app later, you can find it in our SCP Cloud Foundry cockpit. Navigate to your space and select *Applications* in the left menu bar. Click on the name of your app.   Here you will find the route for your application. Click on it to start the “Sentimental Classification App” to test your re-trained model for text classification. |  |

In the next section, you will use the prepared application to test the retrained, newly deployed model.

| Explanation | Screenshot |
| --- | --- |
| 1. Open a new browser window and search for e.g. amazon reviews.   Search for any review / text with a clear sentiment. Copy the text and paste it into our test app. The Classification works best on short texts. Click on *Analyze!* |  |
| 1. You will get the result (classification: positive or negative) visualized as a thumb.   The probability is also shown. |  |

Summary

You have completed the exercise!

You are now able to:

* Use the REST APIs and their Swagger UIs to train your machine learning models.
* Upload and run a custom UI application on the SAP Cloud Platform

## Customizable Image Object Detection

*Note: due to time constraints this exercise is typically only shown as live demo by the trainers.*

Overview

Estimated time: 45 minutes

Objective

In this exercise you learn how to 'fine-tune' the Object Detection machine learning model to a custom use case. You can use this service without being a Machine Learning Expert. The base model is provided by SAP. The Retrain Service adapts the base model to a specific model using training data provided by the customer.

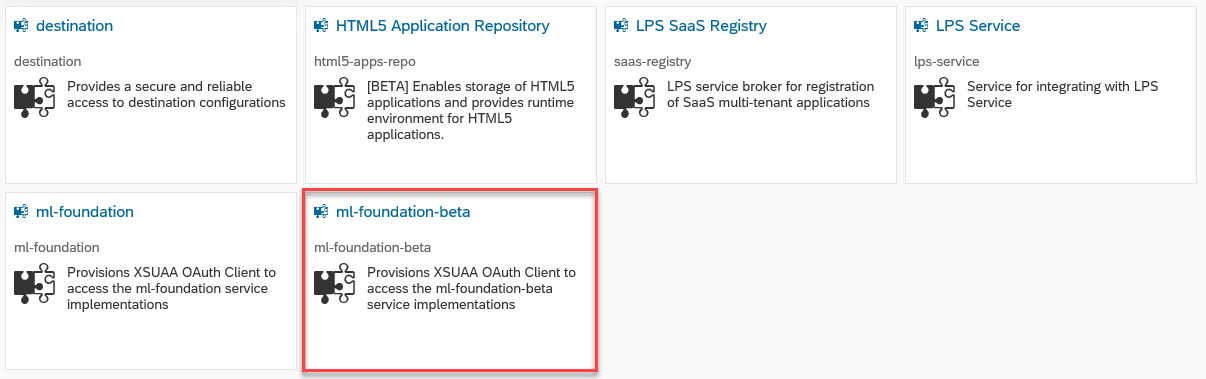
Thus, the detection results for the customer data-set are greatly improved compared to the generic base model.

**Exercise Description** - SAP Leonardo ML Foundation Retrain Service

* *Login to SAP CF using the CF CLI*
* *Create a ML Beta Service Instance and Service Key in SCP Cockpit*
* *Upload the training data.*
* *Perform retraining job using the provided Swagger UI*
* *Deploy the retrained model using the provided Swagger UI*
* *Use the deployed retrained model for inference with the provided Swagger UI*

**Prerequisite**

You need to create a service instance and service key for the ML foundation BETA service in your SAP Cloud Foundry space. Detailed instructions can be found in Appendix 6.2 Setup ML Foundation in an SAP Cloud Platform Trial Account



Also this exercise requires the sapml plugin installed, check Appendix *6.3 Re-train Image Classification with sapml* for a detailed set-up guide.

**Access to SAP Cloud Foundry Space:**

**Username:** ml-train+us-XXX@sap.com

**Password:** <specific to your username>

**API endpoint**: api.cf.us10.hana.ondemand.com

**Subaccount**: ml-train-XXX

**Space:** training

Cloud Foundry Login command:

cf login -a api.cf.us10.hana.ondemand.com -u ml-train+us-XXX@sap.com

Required resources for this exercise:

* racoon-test\_image.jpg

### Training Data (optional)

The data for retraining was already prepared and uploaded to your account. In this exersize you will download the data to your local machine and inspect the general structure.

| Explanation | Screenshot |
| --- | --- |
| 1. Use the following command to download the dataset to your local filesystem   **cf sapml fs get Racoon Racoon**  Note: you may also use the “Retraining App” to have a look into the prepared files. |  |
| 1. The retrainable service takes the following as an input: 2. Set of images 3. Annotations file (.csv format) containing information where the object is located within a given image:  * Image name * Height * Width * Label (ie the object we are trying to detect) * Bounding box (xmin,ymin,xmax,ymax) | C:\Users\i040736\AppData\Local\Temp\msohtmlclip1\02\clip_image003.png  Sample image file (bounding box for better illustration)  Machine generated alternative text: 2  4  6  7  8  g  raccoon- 129  raccoon-3  raccoon-115  raccoon- 101  raccoon-22  raccoon-36  raccoon-37  raccoon-23  raccoon- 100  426  593  576  639  720  426  640  640  640  520  259  960  raccoon  raccoon  raccoon  raccoon  raccoon  raccoon  raccoon  raccoon  raccoon  142  113  276  476  592  *Snapshot of the annotations.csv sample file* |

### Re-train the Image Object Detection Model

In the next section, you will retrain the pre-trained Image Classification Model that is behind the exposed corresponding service. The data for retraining is already uploaded to your filesystem on the cloud data center. You will start a retraining job based on that data. Your re-trained model is created, automatically uploaded to the model repository.

| Explanation | Screenshot |
| --- | --- |
| 1. In the SAP Cloud Platform Cockpit goto the Service Key of your ml-foundation-betaService Instance.   Find the RETRAIN\_OBJECT\_DETECTION\_API\_URL and open the mentioned URL. | {  "clientid": "sb-XXXX-XXXX-XXXX!XXX|ml-foundation-xsuaa-std!XXX",  "appname": " XXXX-XXXX-XXXX!XXX!XXXX|ml-foundation-xsuaa-std!XXXX",  "identityzone": "ml-train-XXX",  "identityzoneid": "XXXXXXXXXXXXXXXX",  "clientsecret": "XXXXXXXXXXXXXXXXX",  "serviceurls": {  …  "IMAGE\_OCR\_URL": "https://mlfproduction-fs-ocr.cfapps.us10.hana.ondemand.com/api/v2/image/ocr",  "RETRAIN\_OBJECT\_DETECTION\_API\_URL": "https://mlfproduction-retrain-od-service.cfapps.us10.hana.ondemand.com",  "SIMILARITY\_SCORING\_URL": "https://mlfproduction-similarity-scoring.cfapps.us10.hana.ondemand.com/api/v2/similarity-scoring",  …  },  "url": "https://ml-train-XXX.authentication.us10.hana.ondemand.com"  } |
| 1. Click the “lock icon” to add the authorization Oauth token generated in the previous exercise.   Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/> |  |
| 1. In the swagger UI find **POST /jobs** operation and click “**Try Out**”.   Edit properties annotationFilePath, imagesDir and modelName to reflect the prepared data set.  You might copy the following json requst:  {  "jobName": "job-training",  "annotationFilePath": "/data/Racoon/annotations.csv",  "imagesDir": "/data/Racoon/images",  "modelName": "object\_detection\_xxx",  "iterations": "1000"  }  **Note**: currently the prefix “/data” is required to address the correct path! This will be obsolete with the next release.  Click *Execute* to submit the retraining job |  |
| 1. The response should show **status code “202”** and print the ID of the newly created job.   Copy the job ID into clipboard. |  |
| 1. You can query the job ID to check the status of training job.   It might take some time for the training container to get allocated and for the training job to run.    Logs of the training job are visible under data\jobs\<name>-<id> or by querying the operation /jobs/<id>/logs. | Machine generated alternative text: GET  / j obs/ { Get status of retraining job with the specified ID.  Information regarding the status of a specific submitted retraining job will be returned.  Pa ram ete rs  Name  id * required  string  (path)  Description  The ID of the job being requested.  Idab40f3-2733-4dac-86f3-1e8ddacdf8fc  Execute  Cancel  Clear  Machine generated alternative text: Server response  Code  200  Details  Response body  "contai ne rld  "fini shTi me  "893e3e315d6fee294f68a68cb899a124f7f6db3b17de1e2e2ee9513a59ßf9e16" ,  " Idab4e ,  message"  name  "job—trai ni ng4" ,  "startTi me  "status  "SUCCEEDED" ,  "submi ssi onTime  "2918-85-31Te6: 19: 58+ee : ee" |

### Deploy the re-trained Object Detection Model

In the next section, you will deploy the model that was created and uploaded after successful training to the model repository. Deploy a model means, a serving container is created that can be used for inference.

| Explanation | Screenshot |
| --- | --- |
| 1. Once the job has succeeded you deploy the model using deployment api. You are still in the swagger UI for object detection and navigate to deployments. Here you can open the POST method to deploy a retrained model. Click on *Try it out*.   Specifying modelname and model version. (model name is the model name used for retraining job) |  |
| 1. Specify the modelName. This is the one that you defined for the re-training you.  Here: object\_detection\_xx.   Version should be 1 as you run this re-training job for the first time.  Enter the bearer token and click on the *Execute* button.  Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/> |  |
| 1. Have a look at the server response. The deployment job gets an ID. Please copy it – we will need it in the next step. |  |
| 1. Query the deployment status using GET /deployments/{id}.   Try it out.  Enter your job id and the bearer token and click on the *Execute* button.  Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/> |  |
| 1. Have a look at the server response. The state should change to SUCCEEDED once model is deployed successfully. |  |

### Test your re-trained Image Object Detection Model

In the next section, you will use the prepared project and adjust it to call the Image Classifier Service with your new model in order to test the retrained, newly deployed model.

|  |  |
| --- | --- |
| 1. In the SAP Cloud Platform Cockpit find he OBJECT\_DETECTION\_API\_URL for your service instance. |  |
| 1. Open the operation /format:image. Enter the model name (object\_detection\_xx), click the “lock icon” to enter your Bearer token and upload an image for which you want to test the classification. *Execute*.   Generate Token Helper App:  <https://generate_ml_token.cfapps.eu10.hana.ondemand.com/> |  |
| 1. The response will visually show bounding boxes and probabilities of corresponding objects detected   Repeat the call usint the root service (“/”) to also retrieve a textual represenation of the results. |  |

Summary

You have completed the exercise!

You are now able to:

* Use the customizable image object detection to
* Retrain your Model
* Deploy your retrained Model
* Use the retrained Model for inference

# Bring Your Own Model (BYOM)

Overview

Estimated time: 45 minutes

Objective

In this exercise you get to experience how easy it is to deploy your machine learning models using SAP Leonardo Machine Learning Foundation services*.* You will learn how to use the SAP ML Endpoints provided in your service key, that you created for your ML Foundation Service instance.

If you have multiple models for the same application you can also upload and choose which version you wish to serve. **Note**: you can not deploy more than one model with the same name.

Afterwards you will deploy your own Cloud Foundry Python app consuming the deployed model to perform inference queries – a so called inference app. For convience reason we prepared a simple SAPUI5 app for you in order to test your deployed model calling the deployed inference app.

**Exercise Description** - SAP Leonardo ML Foundation BYOM service consumption

* *Login to SAP Cloud Platform – Cloud Foundry environment*
* Create MLF Service Instance and Service Key (already done in previous exercise)
* Create a bearer token
* Upload model to ML Foundation Model Repository using Swagger UI
* Deploy your uploaded model using Swagger UI
* Deploy a test app for inference
* Test your new model via this test app

Required resources for this exercise:

* mnist.zip
* Folder template with the inference app
* Folder test images with the test files

**Prerequisite**

First you need to create a service instance and service key for the ML foundation service in your SAP Cloud Foundry space. This is done in the previous exercises.

As the BYOM service is running in the productive Cloud Foundry environment, you will need an Access Token to run the service. Please use the Generate Token helper tool in order to get the OAuth2 token.

**Note**: In the appendix you will also find a description on how to generate the token by yourself with a tool like Postman.

## Generate Access Token

For exercises 4 and 5 we will need a OAuth2 token to access the ML Foundation services in our productive Cloud Foundry environment. We prepared a small helper application. Please use the following link to access this token generation app. You need to be logged in to your SAP Cloud Foundry account, to access it:

<https://generate_ml_token.cfapps.eu10.hana.ondemand.com/>

**Note**: The Generate Token is a tool that we provide during this workshop. It is not part of our standard ML Foundation service offering. In the appendix, you will find a description on how to generate the token by yourself with a tool like Postman.

| Explanation | Screenshot |
| --- | --- |
| 1. A new window comes up where you need to enter the information provided via your service key:   Authentication URL Client-ID Client-Secret  See next step. |  |
| 1. Please go back to your service key and copy and paste the three required values into the Token Generator app. |  |
| 1. Click on |  |
| 1. A token with the Prefix “Bearer” is generated. Use the ***Copy To Clipboard!*** button with which you can copy and paste it.   **Note**: You do not need to copy the token now. But we will reference this section when you need the token to access the ML foundation services. | C:\Users\d037045\AppData\Local\Temp\SNAGHTML1ad14b.PNG |

## Upload and Deploy your Model

In the next section, you will take a prepared TensorFlow model **mnist.zip** and upload it to the ML Foundation model repository. After uploading, you will deploy it which will take a few minutes as the serving containers are instantiated for this model.

**Upload your model**

Required File: mnist.zip

| Explanation | Screenshot |
| --- | --- |
| 1. Navigate in the SCP Cloud Foundry Cockpit to your Service Key. Select the ***Model Repo URL*** and copy it. | xx10 |
| 1. Open a new browser window , paste and open the copied Model Repo URL. A Swagger UI is opened up   (**Note**: it might change in the future). | xx10 |
| 1. Open the third entry > *Model Versions Controller Impl* and select the ***POST URL***. |  |
| 1. This UI provides the function to upload your model.   modelName: enter a name of your choice (here: model\_xx).  File: select the zip file (here: **mnist.zip**) of your model.  **Authorization**: enter your token 🡪 see previous chapter  Click on ***Try it out*** to upload your model. |  |
| 1. In the Response Body you should get a screen as shown on the right side.   Note down the **modelName** and **version** you will need it in the next step. |  |

**Deploy your model**

Required: modelName and version

| Explanation | Screenshot |
| --- | --- |
| 1. Navigate in the SCP Cloud Foundry Cockpit to your Service Key. Select the Deployment URL and copy it. | xx10.hana.ondemand.com |
| 1. Open a new browser window, paste and open the copied Deployment API URL. A Swagger UI is opened up.   We will work with the Model Server Controller. |  |
| 1. Select the POST method.   Enter the followin json string for the modelServerSpecsRequest:  {  "specs": {  "enableHttpEndpoint": false,  "modelRuntimeId": "tf-1.3",  "models": [  {          "modelName": "%MODELNAME%",          "modelVersion": "%MODELVERSION%"  }  ],  "replicas": 1,  "resourcePlanId": "cpu-small\_1\_4"  }  }  Where xxx is your number.  **Authorization**: enter your token    Click on ***Try it out*** to deploy your model. |  |
| 1. You will get a *Response Body* similar to the one on the right side.   The status will be “Pending” in the beginning.  To check the status after a while, we will use another method in the next step. Therefore please copy the modelServerId (“id”). |  |

**Get the information about your deployed Model**

Required: modelServerId

| Explanation | Screenshot |
| --- | --- |
| 1. Within the same Deployment API URL – select the ***Model Server Controller – GET*** method.   /ap/v2/modelServers{modelServerId}  **modelServerId**: Enter the modelServerId you copied before.  **Authorization**: enter your token  Click on ***Try it out*** to get the status of your deployed model. |  |
| 1. Please repeat that step until you get a Response with the deploymentStatus “**SUCCEEDED**”.   Please keep this window open as you need some information in the next step. |  |

## Test your Model

In the next section, you will adjust a prepared inference app with your destinations and credentials. It will then be deployed to you SCP Cloud Foundry space. This step is needed to expose you model via a REST API or URL that can later be used in an application for the inference phase.

**Deploy the prepared test application**

| Explanation | Screenshot |
| --- | --- |
| 1. You need to copy and paste the following information into the manifest.yml of your inference app during the next step. |  |
| 1. Download the zip file  “template.zip” and unpack it in your file system. Open the manifest.yml with a text editor (e.g. Wordpad). |  |
| 1. Enter the following:   name: byom\_app\_xxx  host: byom\_host\_xxx  Where xx is your group name.  Enter your service instance name (ml\_instance\_xx) that you created in SCP cockpit.  The next value ia taken from the step above.  MODEL\_NAME: 🡪 modelName  Save and close this file. |  |
| 1. Open a command prompt and navigate to the folder where the manifest.yml is located.   Login to Cloud Foundry:  cf login -a api.cf.us10.hana.ondemand.com -u ml-train+us-XXX@sap.com  Enter your password. Now you can deploy the application using the command:  cf push | ml-train+us-XXX@sap.com  ml-train-XXX  us10.hana.ondemand.com  us10.hana.ondemand.com |
| 1. Wait until this process has ended. You should see a similar screen where you find the state “running” and the url of your application. | us10.hana.ondemand.com  ml-train+us-XXX@sap.com  ml-train-XXX |
| 1. Take this url with “https://” in front and put it into a new browser window.   The “Test MNIST Model App” should come up. |  |
| 1. In case you want to launch the app later, you can find it in our SCP Cloud Foundry cockpit. Navigate to your space and select Applications in the left menu bar. Click on the link of your app. |  |
| 1. Here you will find the route for your application. Click on it to start the “Test MNIST Model App” to test the model that you uploaded and deployed to MLF (here: mnist). | us10.hana.ondemand.com |

**Test your model**

To test your new model. Run the “Test MNIST Model App” that you deployed in the previous step.

| Explanation | Screenshot |
| --- | --- |
| 1. Open the “Test MNIST Model App”.   Here you can browse you local file system for image files. Select the files we prepared for you in the Exercise 5 subfolder.  Click on Analyze!  Note: This app will always use the latest model and includes the inference API part as well. |  |
| 1. You should get an output that says Result <number> as shown on the screenshot here. |  |

Summary

You have completed the exercise!

You are now able to:

* Use the service key to retrieve the ML Foundation API endpoints
* Upload your Model using the Model API
* Deploy your Model using the Deployment API
* Check the status of your Model Deployment
* Deploy a prepared test app for inference into your CF space
* Test your newly deployed model using the provided test app

# Appendix

## Generate Access Token via Postman

Here is a description on how to generate an access token via a tool like Postman. The token is needed to consume the MLF services in your productive Cloud Foundry environment.

| Explanation | Screenshot |
| --- | --- |
| 1. Navigate to your Service Key and select the ***Authentication URL***.   Note: we need the ***clientid*** and ***clientsecret*** during the next steps. |  |
| 1. Open Postman   Select ***Post*** and paste the ***Authentication URL*** you copied from the previous step. At the end of the URL add the following string **/oauth/token?grant\_type=client\_credentials**  Select the tab *Authorization* and choose the type ***Basic Auth***.  As *Username* paste the ***clientid*** and as *password* the ***clientsecret*** from your service key (see previous step)  Click the ***Send*** button. | us10.hana.ondemand.com |
| 1. Click the Send button.   An access token ist generated. Please copy it and save it for later. | us10.hana.ondemand.com |

## Setup ML Foundation in an SAP Cloud Platform Trial Account

This chapter step-by-step description on how to setup a trial account for SAP ML Foundation Beta Trial.

| Explanation | Screenshot |
| --- | --- |
| 1. Open **Firefox** as Chrome will use SSO for SAP Employees. **Customers can use Chrome.**   Enter the URL for the SAP Cloud Platform Cockpit via [http://cloudplatform.sap.com](http://cloudplatform.sap.com/).  Select the button “Start your free trial”. |  |
| 1. Enter your data in the registration form and provide a valid E-Mail address.   Check the *Terms and Conditions* and click on “Register”.  Note:  You will receive an email. |  |
| 1. An e-mail with a link to activate your account will be in your inbox. |  |
| 1. Check your mails and open the mail with the title “Activate Your Account for SAP Cloud Platform”.   There you will find an activation button. Click on it. |  |
| 1. You will receive a confirmation of your successful registration.  Click on “Continue”. |  |
| 1. The SAP Cloud Platform Cockpit comes up. Here you can now Login to your Trial Account for the first time. |  |
| 1. In parallel you will receive a second mail “Welcome to SAP Cloud Platform”. Here you will find all the details on how to access your acount. |  |
| 1. There are two flavours for using SAP Cloud Platform. Cloud Foundry and Neo.  Please click on Cloud Foundry Trial, as the SAP ML Foundation is provided here. |  |
| 1. A pop-up comes up. Select the Region “Europe (Frankfurt)” and click “OK”. |  |
| 1. After a short processing phase your Global Account is setup with a Subaccount, an Organization and a Space.   Subaccount: trial  Organization: <GlobalAccountname>\_trial  Space: dev  Click on “Go to Space”. |  |
| 1. Within your space go to “Service Marketplace”.   Here you will find a new tile with the name  ml-foundation-trial-beta  Click on that tile. |  |
| 1. Select “Instances” in the menu on the left side.   Click on the button “New Instance” to create an instance for the service ml-foundation-trial-beta. |  |
| 1. The service plan “standard” is given. Click on “Next”.   Note: Within the trial account the service plan is for free. |  |
| 1. You might define some service plan parameters.  Keep it as it is and click “Next”. |  |
| 1. Here you can bind applications to this service instance.  Keep it as it is and choose “Next”. |  |
| 1. Enter an Instance Name of your choice and click “Finish”. |  |
| 1. You will see your instance in the list. Click on it. |  |
| 1. On the left side in the menu choose “Service Keys” and click on “Create Service Key”. |  |
| 1. Give it a name and click on “Save”. |  |
| 1. Your service key is created which is your entry to SAP Machine Learning Foundation.  Congratulation 😊 | C:\Users\d037045\AppData\Local\Temp\SNAGHTML21ba9a0.PNG |

## Re-train Image Classification with sapml

Overview

Estimated time: 30 minutes

Objective

In this exercise you learn how to 'fine-tune' a generic machine learning model to a custom use case. You can use this service without being a Machine Learning Expert. The base model is provided by SAP. The Retrain Service adapts the base model to a specific model using training data provided by the customer.

Thus, the classification results for the customer data-set are greatly improved compared to the generic base model.

**Exercise Description** - SAP Leonardo ML Foundation Retrain Service

* *Install SAP CF CLI (done in chapter 3)*
* *Login to SAP CF using the SAP CF CLI*
* *Create Retrain Service Instance and Service Key (already done in previous exercise)*
* *Install [SAPML CLI] plugin*
* *Login to access ML Foundation services*
* *Upload the training data. (Note: the training data is already uploaded for this exercise)*
* *Perform retraining process using the sapml command line tool*
* *Deploy the retrained model using the sapml command line tool*
* *Use the deployed retrained model for inference (with a prepared SAPUI5 app)*

**Prerequisite**

You need to create a service instance and service key for the ML foundation service in your SAP Cloud Foundry space. **Please complete exercise 0** before you proceed with this exercise.

For this exercise, you will need the **SAP CF CLI** and the SAPML CLI plugin (this is done during this exercise here).

The data for retraining is already uploaded to the file system associated with your Cloud foundry ML Foundation service account. This is a step that you need to perform as well in a productive environment or POC. We skipped it during this workshop as it would take some time.

As the retraining service is running in the productive Cloud Foundry environment, you will need an Access Token to run the service. Please use the Generate Token helper tool in order to get the OAuth2 token.

Note: In the appendix, you will also find a description on how to generate the token by yourself with a tool like Postman.

**Access to SAP Cloud Foundry Space:**

**Username:** ml-train+us-XXX@sap.com

**Password:** <specific to your username>

**API endpoint**: api.cf.us10.hana.ondemand.com

**Subaccount**: ml-train-XXX

**Space:** training

Cloud Foundry Login command:

cf login -a api.cf.us10.hana.ondemand.com -u ml-train+us-XXX@sap.com

Required resources for this exercise:

* sapmlcli Plugin for CF CLI (Windows, Mac OS or Linux version)
* retrain.json
* Sample images: BMW.jpg, SAP\_perturbed.jpg, SAP\_pertubed\_5.jpg, SAP\_WDF19.jpg

Retrain and Deploy the retrained Image Classifier Model

In the next section, you will retrain the pre-trained Image Classification Model that is behind the exposed corresponding service. The data for retraining is already uploaded to your filesystem on the cloud data center. You will learn how you can initialize the data for your SAP Cloud Foundry account to see its structure. You will start a retraining job based on that data. Your re-trained model is created, automatically uploaded to the model repository and you will deploy it for inference.

| Explanation | Screenshot |
| --- | --- |
| 1. Connect to your remote host or make sure, that you work with an environment where the SAP CF CLI is installed. Open a command prompt (cmd.exe) and enter cf to check if the SAP CF CLI is available. |  |
| 2. Login to SAP Cloud Platform.  Enter:  cf login -a api.cf.us10.hana.ondemand.com -u ml-train+us-XXX@sap.com  Give the password mentioned in **Cloud Foundry** **access** | C:\Users\student> cf login -a [https://api.cf.us10.hana.ondemand.com](https://api.cf.eu10.hana.ondemand.com) -u ml-train+us-XXX@sap.com  API endpoint: https://api.cf.us10.hana.ondemand.com  Password>  Authenticating...  OK  Targeted org ml-train-XXX  Targeted space training  API endpoint: https://api.cf.us10.hana.ondemand.com (API version: 2.96.0)  User: ml-train+us-XXX@sap.com  Org: ml-train-XXX  Space: training  C:\Users\student>:~$ |
| 1. Enter the command:   cf service-key <instance name> <service key>  to display your service key and make sure that everything is available. | ml-train+XX@sap.com |
| 1. Dowload the SAPML CF plugin from:   [https://tools.hana.ondemand.com/#mlfoundation](https://tools.hana.ondemand.com/%23mlfoundation) |  |
| 1. Install SAPML CF CLI plugin   Run your command prompt (e.g. on windows: cmd.exe):  cf install-plugin -f **<your Folder>**\sap\_ml\_cli\_windows.exe  **(you downloaded the Exercises locally 🡪 use your own path)**  For Linux, Unix use the sap\_ml\_cli\_linux/mac\_os.bin version  This CLI plugin is an extension of standard CF plugin. We have added a subcommand called `sapml` to easily interact with our ML Foundation APIs. | C:\Users\student> cf install-plugin <your folder>\sap\_ml\_cli\_windows.exe  \*\*Attention: Plugins are binaries written by potentially untrusted authors. Install and use plugins at your own risk.\*\*  Do you want to install the plugin <your path>\sapmlcli.exe?> yes  Installing plugin sapmlcli.exe...  OK  Plugin SAPML v0.2.0 successfully installed.  C:\Users\student> |
| 1. Login to ML Foundation | NOTE: You are automatically logged in to use ML Foundation |
| 1. Enter the command:   cf sapml config get  You will see that some values are not set, after you installed the sapml plugin. In case you used it before, make sure, that the values point to the right API endpoints. See next step. | C:\Users\Student\cf sapml config get  Key Value  job\_api  ml\_foundation\_service\_name ml-foundation  retraining\_text\_api  retraining\_image\_api  deployment\_api  model\_repo\_api  auth\_server |
| 1. Set the correct values for your sapml configuration. You will need the values of your Service Key (see chapter 3.1 Install Cloud Foundry Command Line Interface (CF CLI)   To work with the Cloud Foundry Command Line Interface (CF CLI) you first need to install it:   * Go to https://help.sap.com/viewer/65de2977205c403bbc107264b8eccf4b/Cloud/en-US/856119883b8c4c97b6a766cc6a09b48c.html  and choose ***Download and Install the Cloud Foundry Command Line Interface*** * Read the procedure for ***Windows*** respectively ***Mac OS*** and do the installation   + For Microsoft Windows, download the Windows 64-bit installer and start it   + For Mac OS, download the PKG file and start the installer  1. Create ML Foundation Service Instance & Service Key:   cf sapml config set auth\_server <your authentication URL from the service key>  cf sapml config set job\_api https://training.prod.eu-central-1.aws.ml.hana.ondemand.com  cf sapml config set retraining\_image\_api [https://mlfproduction-retrain-image-api.cfapps.us10.hana.ondemand.com/api/v2/image/retraining](https://mlfproduction-retrain-image-api.cfappsus10.hana.ondemand.com/api/v2/image/retraining)  cf sapml config set retraining\_text\_api <https://mlfproduction-retrain-text-linear-api.cfapps.us10.hana.ondemand.com/api/v2/text/retraining>  Enter “cf sapml config get” to check if everything is set correctly. | C:\Users\Student\cf sapml config set auth\_server https://ml-train-XXX.authentication.us10.hana.ondemand.com  C:\Users\Student\cf sapml config set job\_api https://training.prod.eu-central-1.aws.ml.hana.ondemand.com  C:\Users\Student\cf sapml config set retraining\_api https://mlfproduction-retrain-image-api.cfapps.us10.hana.ondemand.com/api/v2/image/retraining |
| 6. Initialize the cloud filesystem:  cf sapml fs init | C:\>cf sapml fs init  Attempting automatic login using cloud foundry service "ml-foundation" ...  Login success in service "ml-foundation".  Initializing. Please wait...  Initialization finished successfully |
| 7. List the root directory  cf sapml fs list  Display the subfolders of the already available “Brands” training data.  cf sapml fs list Brands/  Display the training categories.  cf sapml fs list Brands/training/ | C:\Users\student>cf sapml fs list  Brands/  C:\Users\student>cf sapml fs list Brands/  Brands/test/  Brands/training/  Brands/validation/  C:\Users\student>cf sapml fs list Brands/training/  Brands/training/Adidas/  Brands/training/Apple/  Brands/training/BMW/  Brands/training/Coca-Cola/  Brands/training/SAP/ |
| 7. Retrain configuration: Edit the file “retrain.json”.  You will now change the configuration so that it uses the prepared data.  You need to specify the data set name and a model name. As our folder structure has the root name “Brands”, we choose the value “Brands” for our dataset.  Give it a new model name of your choice and save it (e.g. brands-xxx).  You can add optional retrain parameters (e.g. learning rate). | {  "dataset": "Brands",  "modelName": "your-model-name"  "learningRate": 0.001  } |
| 8. Start the retrain process using sapmlcli:  cf sapml retraining job\_submit [PATH\_TO\_JSON] -m image  Example:  cf sapml retraining job\_submit C:\Users\student\retrain.json -m image  Check the job status:  cf sapml retraining jobs -m image | C:\Users\student>cf sapml retraining job\_submit retrain.json -m image  Job submitted successfully with id brands-2017-12-13t1612z.  C:\Users\student>cf sapml retraining jobs -m image  ID STATUS  brands-2017-12-13t1612z Succeeded |
| 9. Have a look at the file system structure of your remote fs with the command:  cf sapml fs list  You will see a new folder with the job-name “brands-<date><id>”. Inside this folder a log file is written.  Download the retrain log:  cf sapml fs get [REMOTE\_PATH] [LOCAL\_PATH]  Example:  cf sapml fs get brands-*your-job-name*/retraining.log myretrain.log | C:\Users\student>cf sapml fs get brands-2017-12-13t1612z/retraining.log myretrain.log  Calculating download size...  6.86 KiB / 6.86 KiB [========================================================================] 100.00% 326.80 KiB/s 0s |
| 10. Display the retrain log:  You can open it in a text editor or use e.g. the ‘more’ command.  Note: the model version corresponds to the retrained version.  It is incremented for every retrain run when using the same model name.  Wait until you get the status *Succeeded* for your retraining job. | C:\Users\student>more myretrain.log  Scanning dataset Brands ...  Dataset used: Brands  Dataset has labels: ['Adidas', 'Apple', 'BMW', 'Coca-Cola', 'SAP']  1270 images are used for training  323 images are used for validation  323 images are used for test  …  …  ##########################################  ########### Retraining Summary ###########  ##########################################  Job id: brands-2017-12-13t1612z  Training batch size : 64  Learning rate : 0.001000  Total retraining epochs : 100  Retraining is stopped after 10 consecutive epochs which show no improvement in accurracy.  Epoch with best accuracy : 9  Best validation accuracy : 0.965944  Final test accuracy is : 0.972136  The exported model will predict top 5 classifications  Retraining started at: 2017-12-13 16:12:30  Retraining ended at: 2017-12-13 16:15:54  …  Retraining lasted: 0:03:23.589231  Model is uploaded to repository with  name brands-49 and version 1. |
| 11. Deploy the retrained model:  cf sapml retraining model\_deploy [MODEL\_NAME] [MODEL\_VERSION] -m image  Example:  cf sapml retraining model\_deploy brands*-49* 1 -m image | C:\>cf sapml retraining model\_deploy brands-49 1 -m image  Deployment request submitted with id 283b6cb8-3446-41f3-8677-32e7f6abda88.  C:\>cf sapml retraining model\_deployments -m image  ID MODEL NAME MODEL VERSION STATUS  283… brands-49 1 PENDING  URL is not yet reachable |
| 12. Check the deployment status:  (it takes a couple of minutes until the model container with the retrained model is up and running)  cf sapml retraining model\_deployments -m image  Deployment status (SUCCEEDED). | C:\>cf sapml retraining model\_deployments -m image  ID MODEL NAME MODEL VERSION STATUS  283… brands-49 1 SUCCEEDED |