Tooling for non-AI experts - Watson Studio AutoAI

Get experience with IBM Watson Studio's no-code *AutoAI* tool by creating a binary classification Machine-Learning model to evaluate the risk that a customer might leave your service.

Duration: 30 minutes

In this tutorial, you will use IBM Watson Studio's AutoAI to train and deploy a machine-learning model within a low-code/no-code paradigm.

Introduction

Objectives

In this hands-on lab, you will learn how to create, train and deploy a Machine Learning model with a no-code/low-code paradigm, leveraging Watson Studio's AutoAI tool.

Applying supervised Machine Learning

In supervised Machine Learning, a model is trained from historical data. The training phase is fed with records representing past observations of a dataset's behavior, and a model representing the beahvior of this dataset is produced.

Telco *Customer Churn* use-case

The use case that we will tackle here is to predict whether a customer is likely to switch telephony operator or not. We have at our disposal a dataset, `customer_churn`, which represents past observations of customers' `CHURN` behavior, alongside defining characteristics of each customer, such as demographics data on age, gender, number of children, and business domain data pertaining to its characteristics as a customer, such as plan, payment method, average usage, …

Once the model will have been trained, it can be used to predict the `CHURN` indicator for new customer records. This can then be used for example to orient actions to be taken when the customer is in contact with a call center, or to drive a marketing customer retention campaign.

Predictive Model setup

The value that we want to predict is represented by a string which can take two values or classes, `T` or `F`. This means that we are facing a *Binary Classification* type of ML problem.
There are many possible implementations of algorithms to solve Classification problems, and Watson AutoAI will help us determine the implementation which has the best accuracy for the training set.
Without going into details, each type of algorithm has several accuracy measurement indicators which can be chosen depending on the overall 'shape' and constitution of the dataset, as well as the intended use of the prediction.
For binary classification, a standard metric is called AUC-ROC (for Area Under Curve-Receiver Operator Characteristics), and measures how well a model is able to predict both Positives and Negatives. Other indicators sur as AUC-PR (AUC-Precision Recall) measures how well a model is able to identify Positives, even if Negatives prediction is less accurate.

Hands-on Lab overview

The instruction below will guide you in achieving the following tasks:

- Load a data set into the project
- Use IBM Watson Studio AutoAI to train, test, and evaluate a machine-learning model
- Deploy the trained model

* Use the deployed model to generate predictions on a new dataset

Hands-on Lab steps

> Note: setup should already be done so you can skip to step [A]. Note that if you already have a project created, you can reuse it, it is not required to recreate a new one.

[0] Setting up Watson Studio within IBM Cloud

This section is a one-time setup to add the *Lite* version of Watson Studio to your IBM CLoud account. Watson Studio Lite version is fully functional but has some capacity limits, which will not be exhausted by running this lab.

Watson Studio is an IBM Cloud service, so in addition to the IBM Cloud account setup, you will need to create the Watson Studio instance. In addition, Watson Studio makes use of additional data and AI related services from the IBM Cloud platform, so we will create some artifacts for use within Watson Studio at runtime:

1. Create a Watson Studio service instance
2. Create a Watson Studio Project for the workshop.
3. Provision a set of additional services
4. Load data files into the project as Data Assets

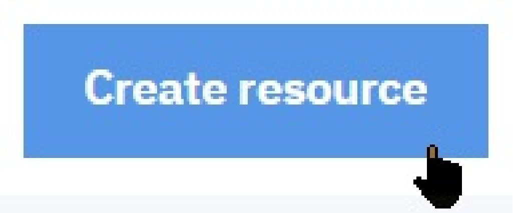
[0.1]. Accessing IBM Cloud

This section assumes that the Master Class pre-work has been completed and that you have access and are logged into an IBM Cloud account at https://cloud.ibm.com.
If not, follow instructions in SophIA_WatsonStudio_Setup_v9.pdf

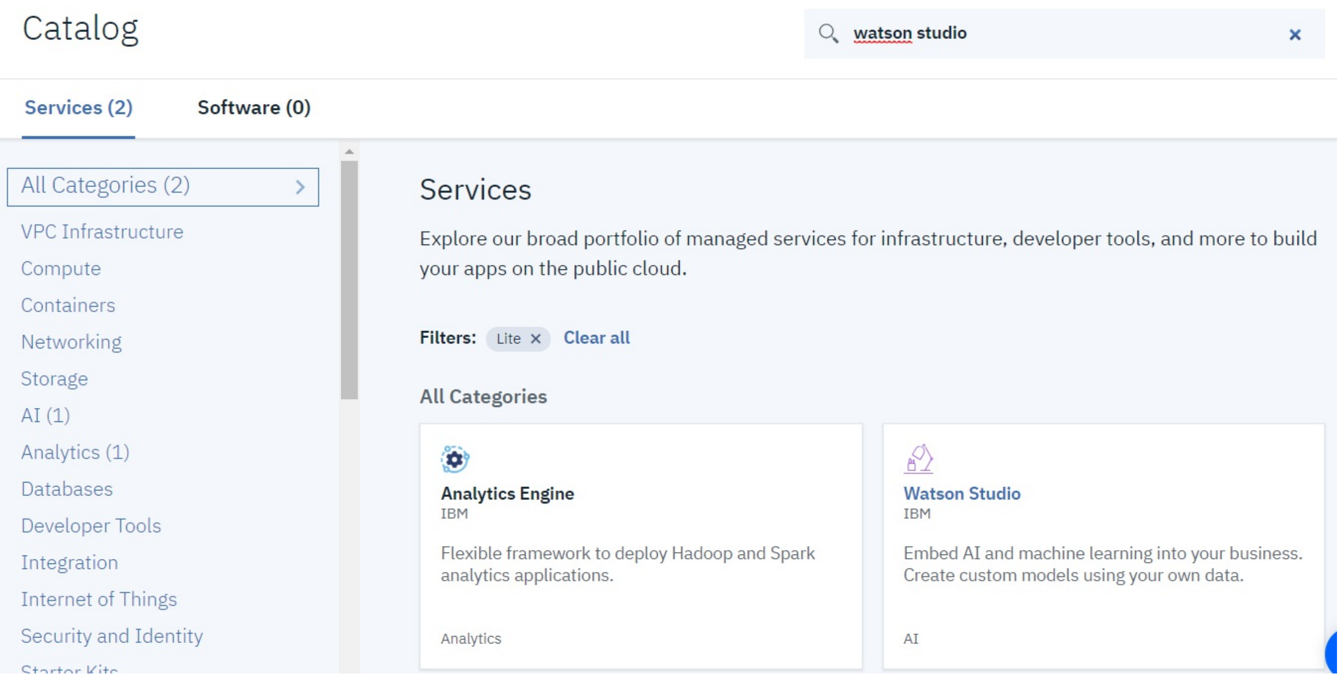
[0.2]. Creating a Watson Studio instance

From IBM Cloud, we will instanciate a Watson Studio service, as the anchor for the toolset within IBM Cloud. Note that this is a one-time setup, only one instance of Watson Studio per region needs to be created.

1. Log-in to you IBM Cloud account's dashboard (https://cloud.ibm.com)



2. Click the `[Create Resource]` button at the top right
3. In the search filter field, type `watson studio`. This should reveal the lite services having the `watson studio` word in their description.



Services

Explore our broad portfolio of managed services for infrastructure, developer tools, and more to build your apps on the public cloud.

Filters: Lite × Clear all

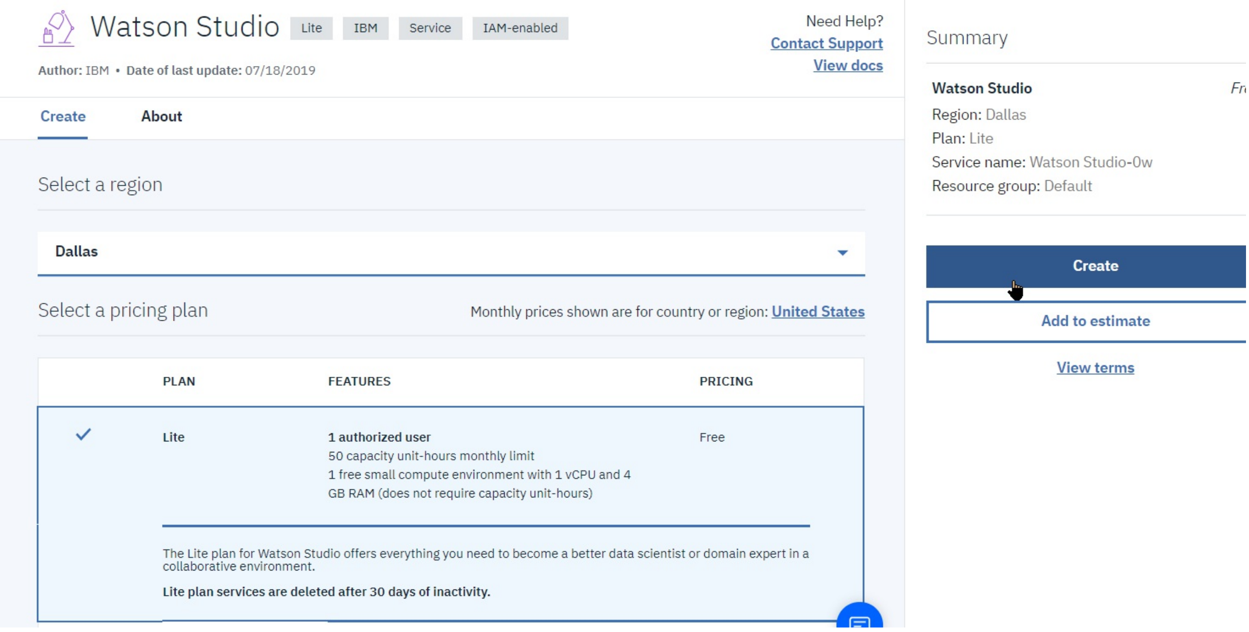
All Categories

Analytics Engine
IBM
Flexible framework to deploy Hadoop and Spark analytics applications.
Analytics

Watson Studio
IBM
Embed AI and machine learning into your business. Create custom models using your own data.
AI

and click the `Watson Studio` tile.

> NOTE: Make sure to use `Watson Studio`, and *not* `Knowledge Studio`
>
> 4. You are taken to the service creation page. Although it is possible to create an instance of

> Watson Studio in either `Dallas`, `London` or Frankfurt data centers, it will be more convenient to use `Dallas` for the purpose of this masterclass. You can change the service name suffix or keep the suggested name. Keep the `Lite` service plan and click the `[Create]` button.
>
> 
>
> Watson Studio Lite IBM Service IAM-enabled
>
> Author: IBM • Date of last update: 07/18/2019
>
> Need Help?
> Contact Support
> View docs
>
> Select a region
>
> Dallas
>
> Select a pricing plan
>
> Monthly prices shown are for country or region: United States
>
> | | PLAN | FEATURES | PRICING |
> |---|---|---|---|
> | ✓ | Lite | **1 authorized user**
50 capacity unit-hours monthly limit
1 free small compute environment with 1 vCPU and 4 GB RAM (does not require capacity unit-hours) | Free |
>
> The Lite plan for Watson Studio offers everything you need to become a better data scientist or domain expert in a collaborative environment.
>
> **Lite plan services are deleted after 30 days of inactivity.**
>
> Summary
>
> **Watson Studio** Fr
> Region: Dallas
> Plan: Lite
> Service name: Watson Studio-0w
> Resource group: Default
>
>

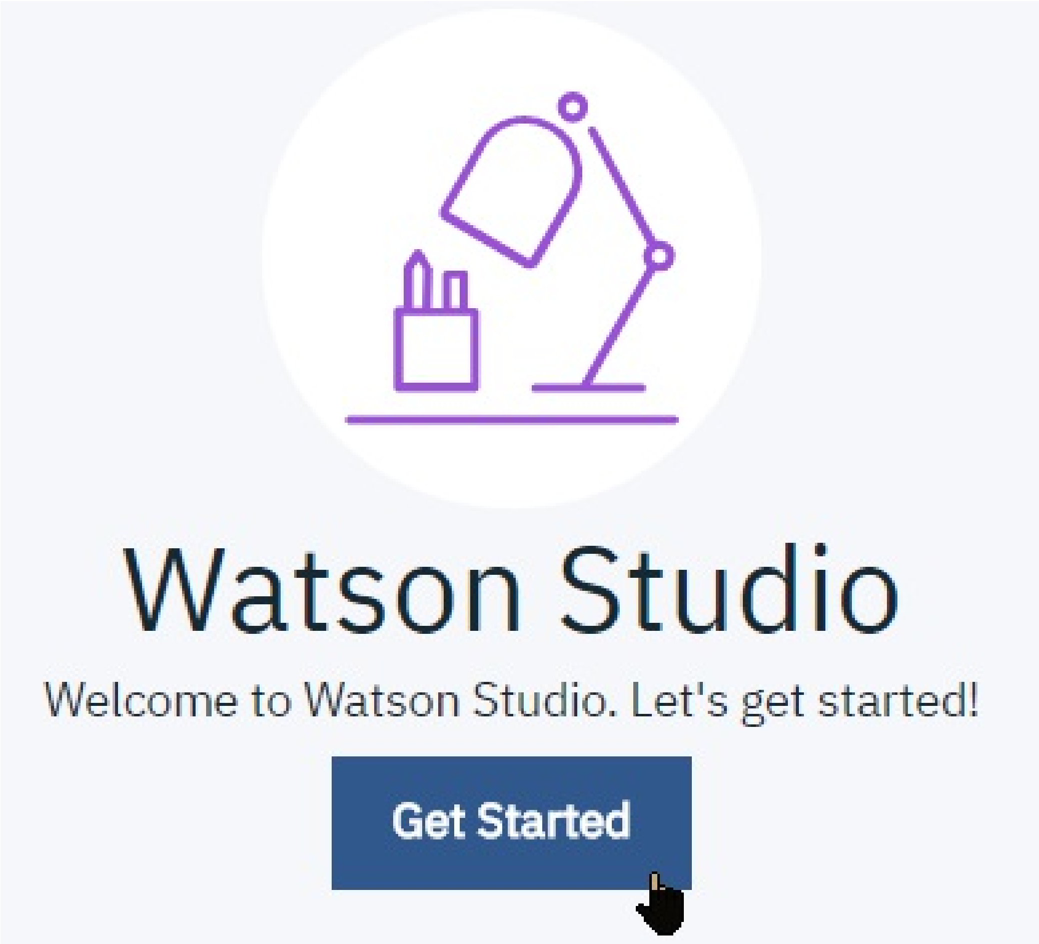
NOTE: In the rest of the labs, if you created your Waston Studio instance in the `Dallas` data center, you will use the plain URLs without prefix, e.g. `dataplatform.ibm.com`, but if you created in the `London` data center, you will need to use the `eu-gb` URLs, e.g. `eu-gb.dataplatform.ibm.com`, or `eu-de.dataplatform.ibm.com` for `Frankfurt` center.

[A] Building a Predictive Model using AutoAI

[A.1] Project setup

Now that we have put in place the infrastructure to work with Data & AI, we can start creating a project for a specific data handling project.

1. If not already signed-in, login to your Watson Studio environment within IBM Data Platform. For this, go back to the IBM Cloud dashboard, select the `Watson Studio` service instance, and click the '[Get

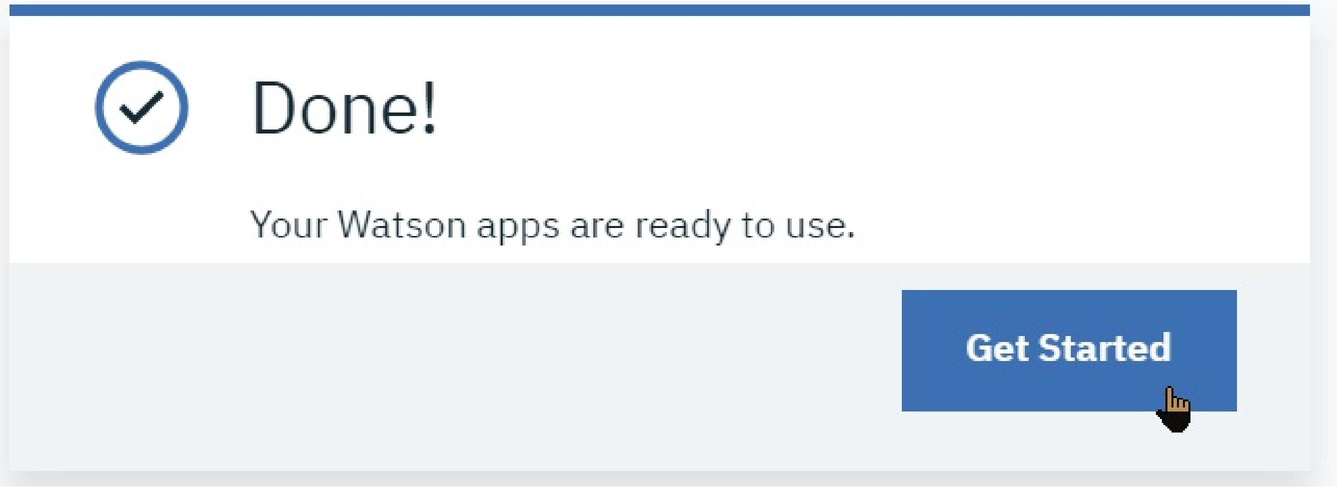


Watson Studio

Welcome to Watson Studio. Let's get started!

Started]' button .

The first time you start the Watson Studio UI, you will be asked to confirm some details, click `[Get Started]`



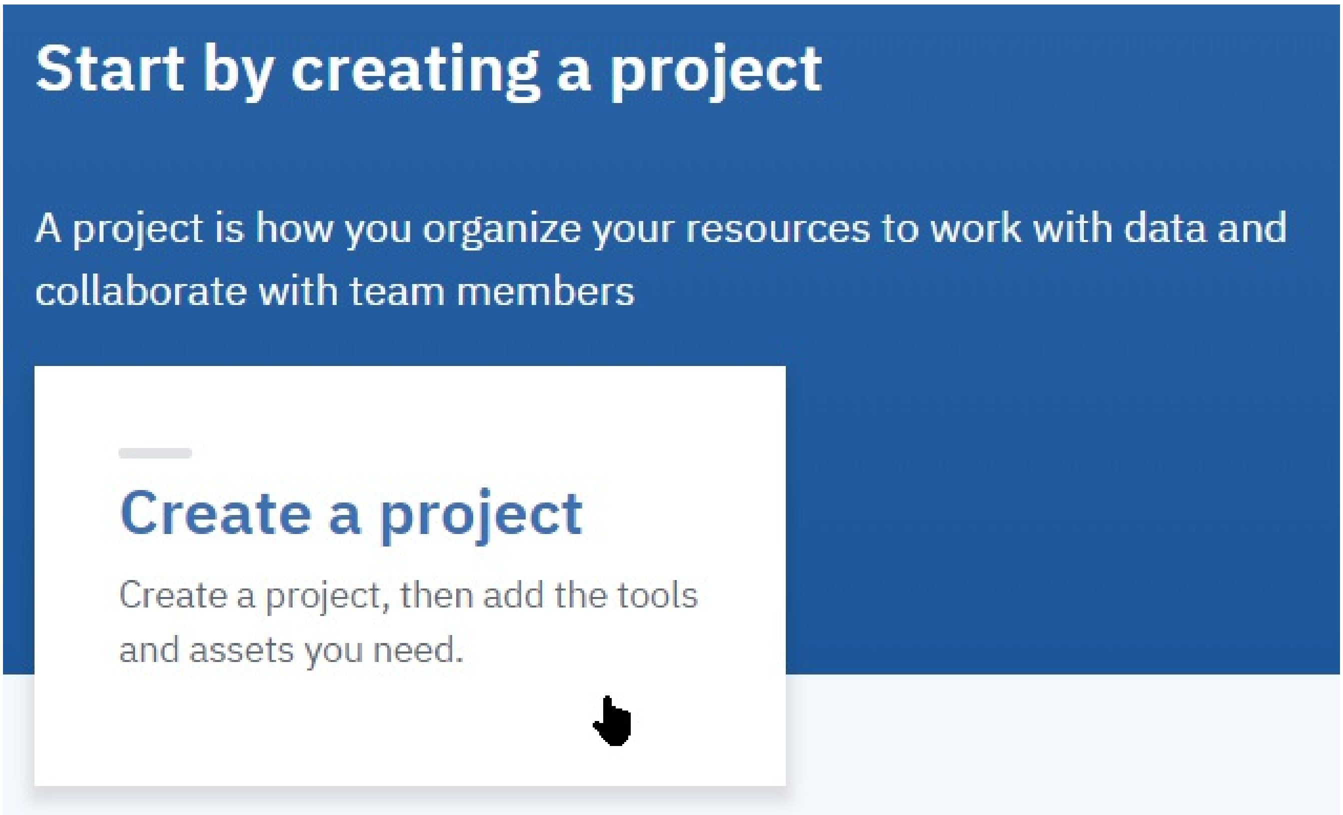
Done!

Your Watson apps are ready to use.

Note that you can also go directly to the service's Cloud Web UI using the URL for the dat acenter where the service has been created, either

- https://dataplatform.ibm.com/projects?context=analytics for 'Dallas'
- https://eu-gb.dataplatform.ibm.com/projects?context=analytics for 'London'
- https://eu-de.dataplatform.ibm.com/projects?context=analytics for 'Frankfurt'

Create a new project using the `Create a Project` button tile



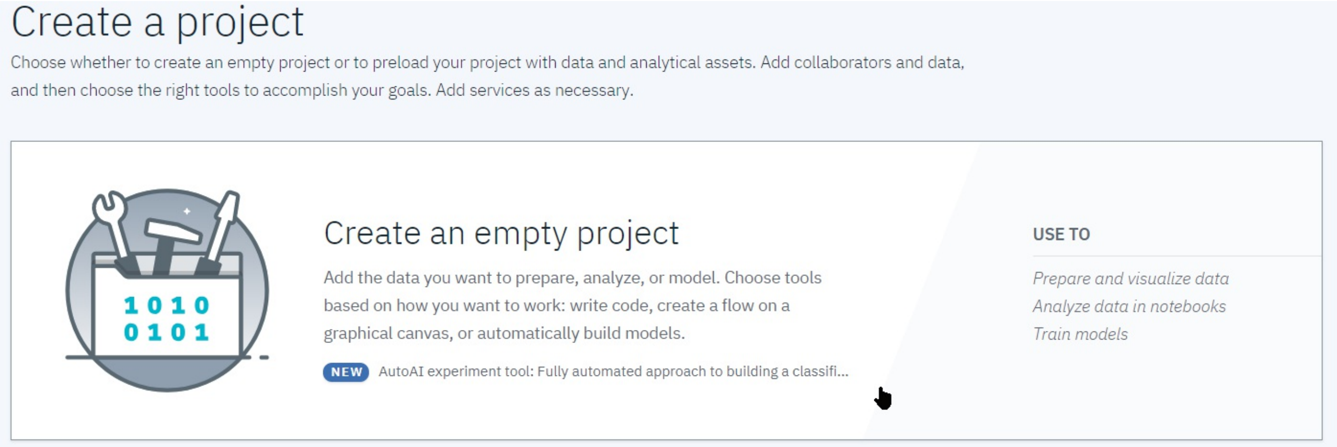
Start by creating a project

A project is how you organize your resources to work with data and collaborate with team members

Create a project

Create a project, then add the tools and assets you need.

Then select a `Create an empty project` configuration.



Create a project

Choose whether to create an empty project or to preload your project with data and analytical assets. Add collaborators and data, and then choose the right tools to accomplish your goals. Add services as necessary.

Create an empty project

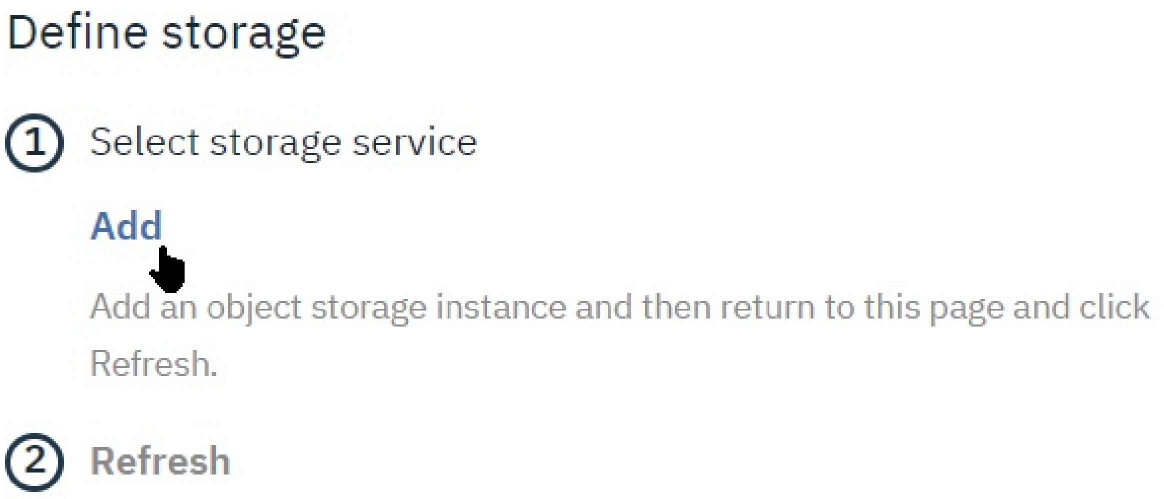
Add the data you want to prepare, analyze, or model. Choose tools based on how you want to work: write code, create a flow on a graphical canvas, or automatically build models.

NEW AutoAI experiment tool: Fully automated approach to building a classifi...

USE TO

Prepare and visualize data
Analyze data in notebooks
Train models

1. Name this new project e.g. `SophIA2019`. Watson Studio stores its file-like artifacts into an instance of `Cloud Object Storage`, we will create a COS service instance at this stage:
   i. Click on `[Add]` in the Define Storage section



Define storage

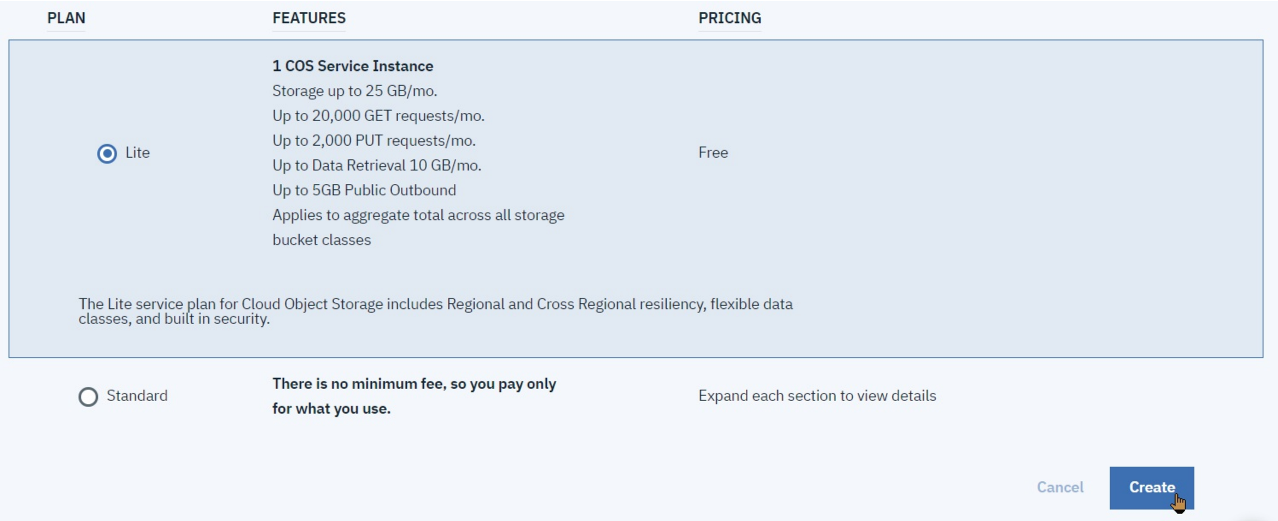
① Select storage service

Add

Add an object storage instance and then return to this page and click Refresh.

② Refresh

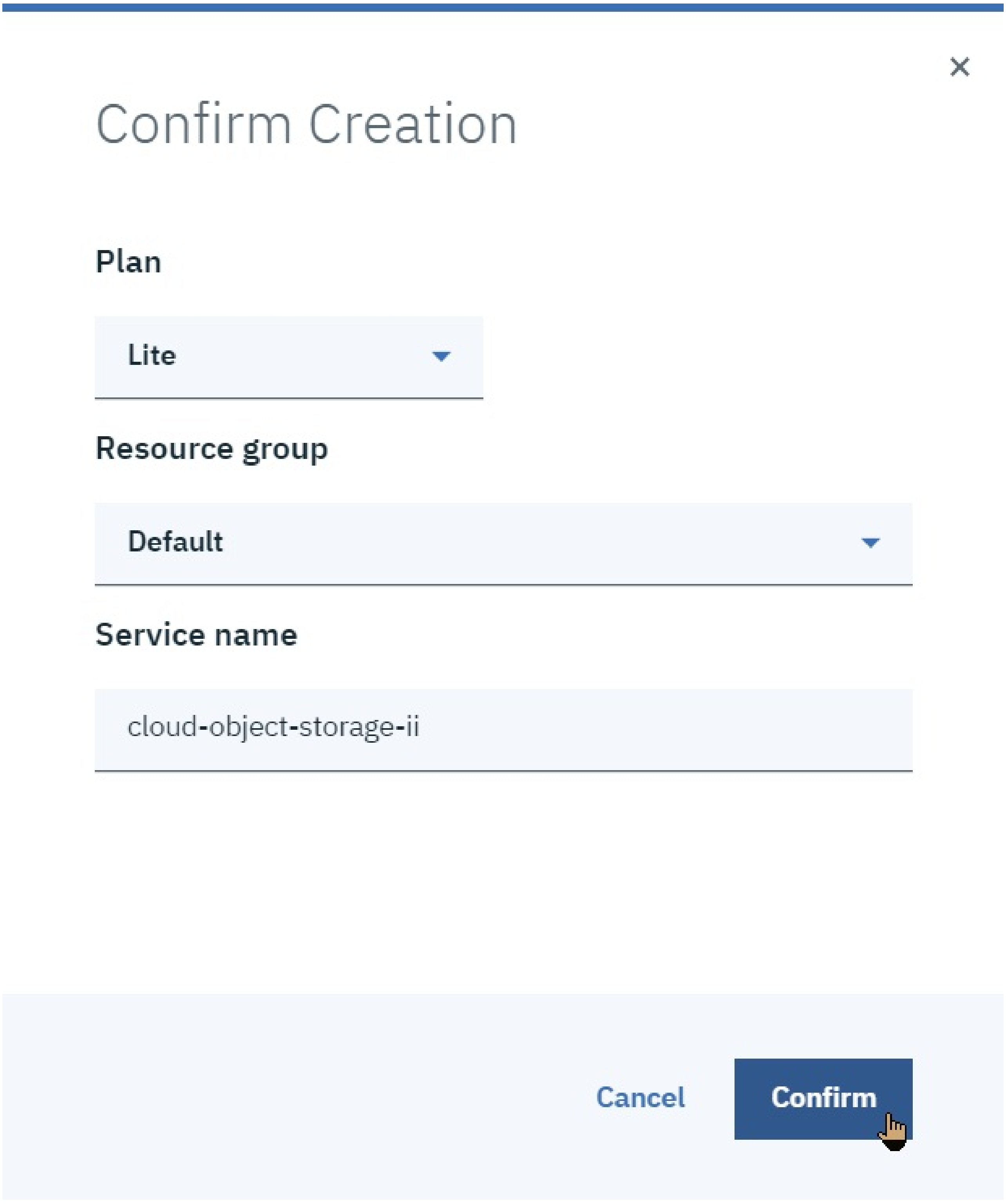
   ii. Select the Lite Plan



PLAN	FEATURES	PRICING
● Lite	**1 COS Service Instance** Storage up to 25 GB/mo. Up to 20,000 GET requests/mo. Up to 2,000 PUT requests/mo. Up to Data Retrieval 10 GB/mo. Up to 5GB Public Outbound Applies to aggregate total across all storage bucket classes	Free
The Lite service plan for Cloud Object Storage includes Regional and Cross Regional resiliency, flexible data classes, and built in security.		
○ Standard	**There is no minimum fee, so you pay only for what you use.**	Expand each section to view details

and click `[Create]`

iii. Accept defaults



Confirm Creation

Plan

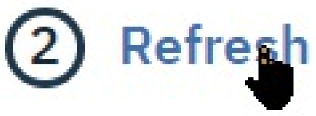
Lite

Resource group

Default

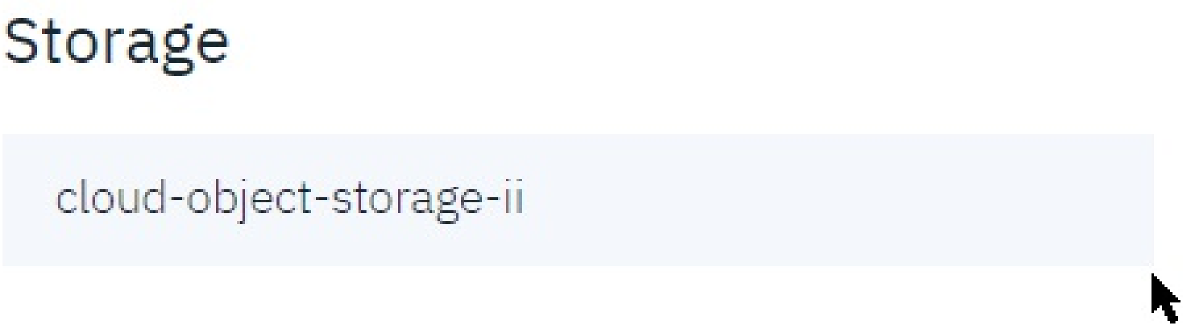
Service name

cloud-object-storage-ii



iv. Back to the Project creation page, select Refresh,

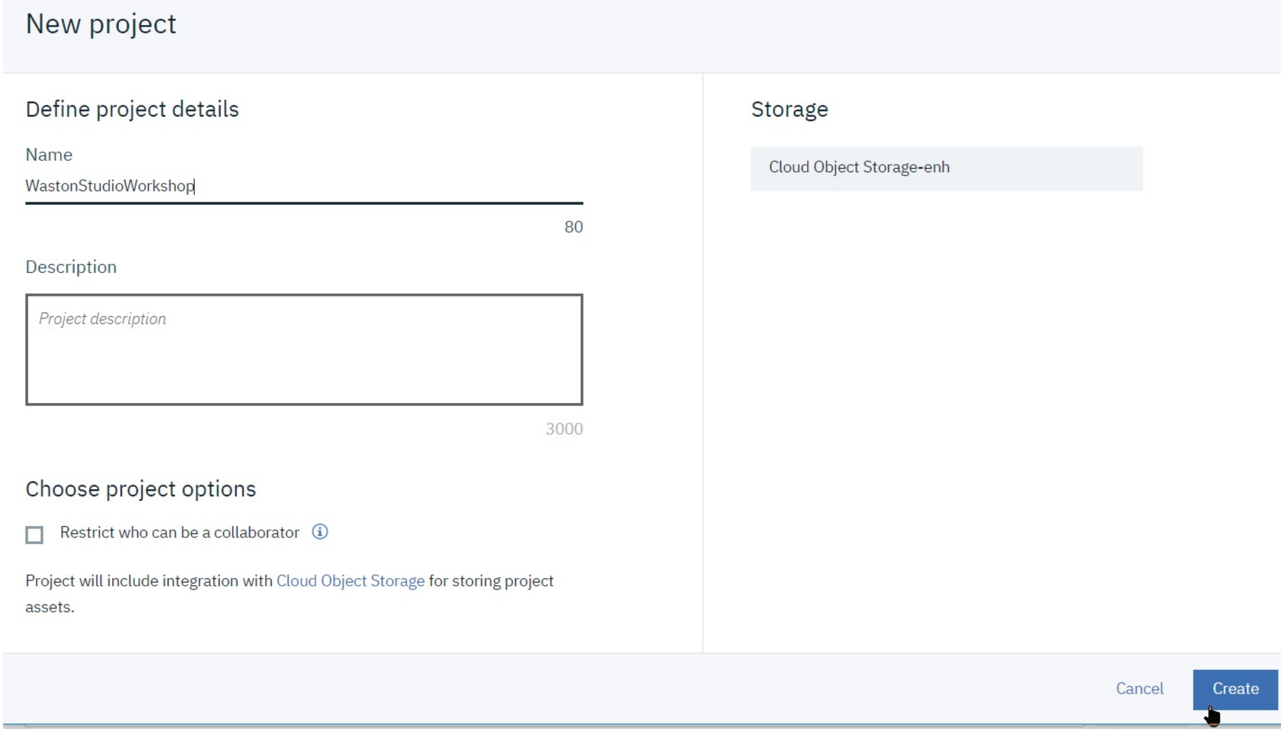
v. The newly created Object Storage service instance should be preselected



Storage

cloud-object-storage-ii

vi. Finally, click Create for the project:



New project

Define project details

Name

WastonStudioWorkshop

80

Description

Project description

3000

Choose project options

□ Restrict who can be a collaborator

Project will include integration with Cloud Object Storage for storing project assets.

Storage

Cloud Object Storage-enh

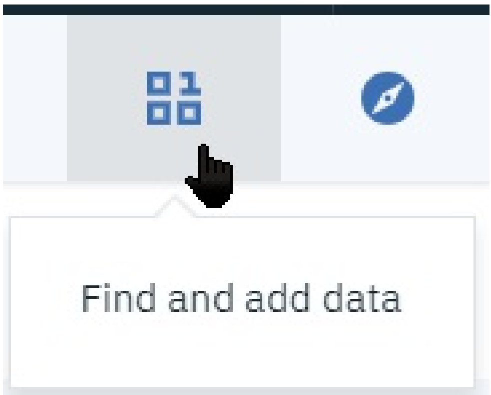
Note that COS instance needs to be created only once, it will hold projects' artifacts in separate buckets for each.

You may think of Cloud Object Storage as the file system for the Cloud.

[A.2] Data Preparation for the Training phase

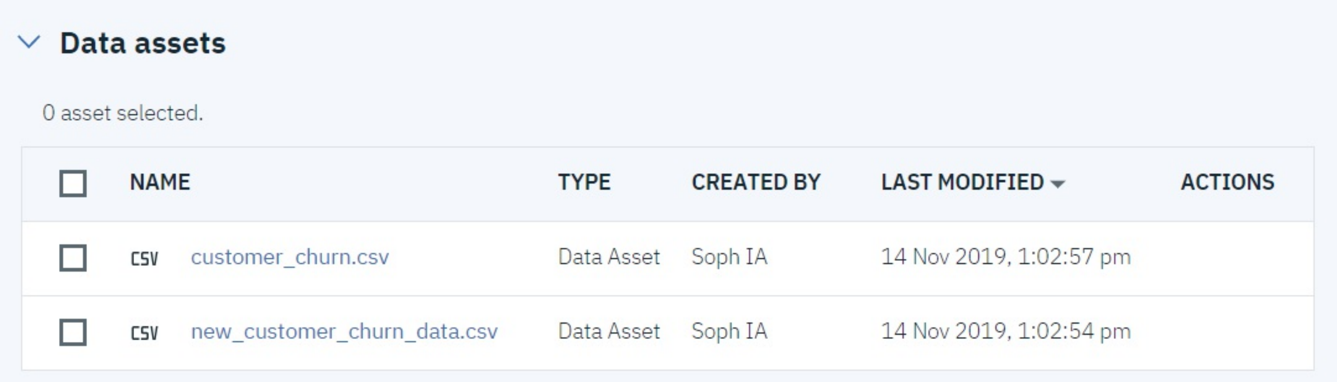
In this section, we will prepare the data to be used for model training and verification.
The data files have been stored in the Box folder, and should be downloaded to your laptop first.

1. In your project, switch to the `Assets` tab.
2. In this tutorial, you work with a data sets stored as project artifacts. Click the Find and add Data icon



which looks like a `10 01` button . It will open the file management sidebar.

3. From the Load tab, Click Browse to select from your local file system. Navigate to the lab files folder and select both `customer_churn.csv` and `new_customer_churn_data.csv` files and click Open.
4. The two files will now be listed in the Data assets section:



Data assets

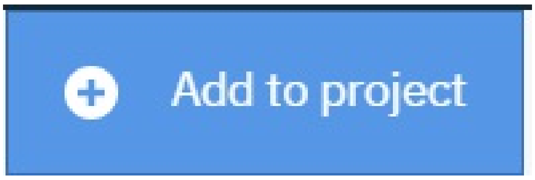
0 asset selected.

□	NAME	TYPE	CREATED BY	LAST MODIFIED	ACTIONS
□	CSV customer_churn.csv	Data Asset	Soph IA	14 Nov 2019, 1:02:57 pm	
□	CSV new_customer_churn_data.csv	Data Asset	Soph IA	14 Nov 2019, 1:02:54 pm	

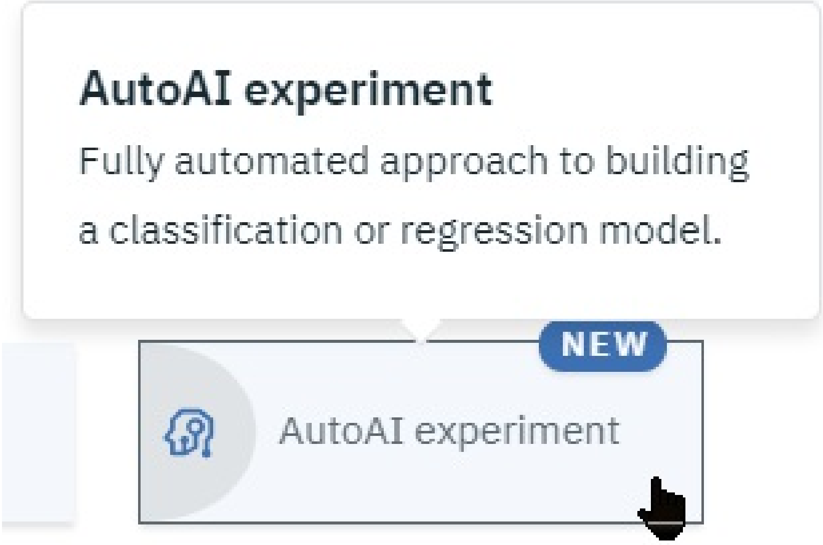
Alternatively, you can drag and drop a file directly into the sidebar.
The file is added to your local data sets in your project.

[A.3] Predictive Model training with AutoAI

We will now create a model by using the IBM Watson Studio's AutoAI low-code model builder.



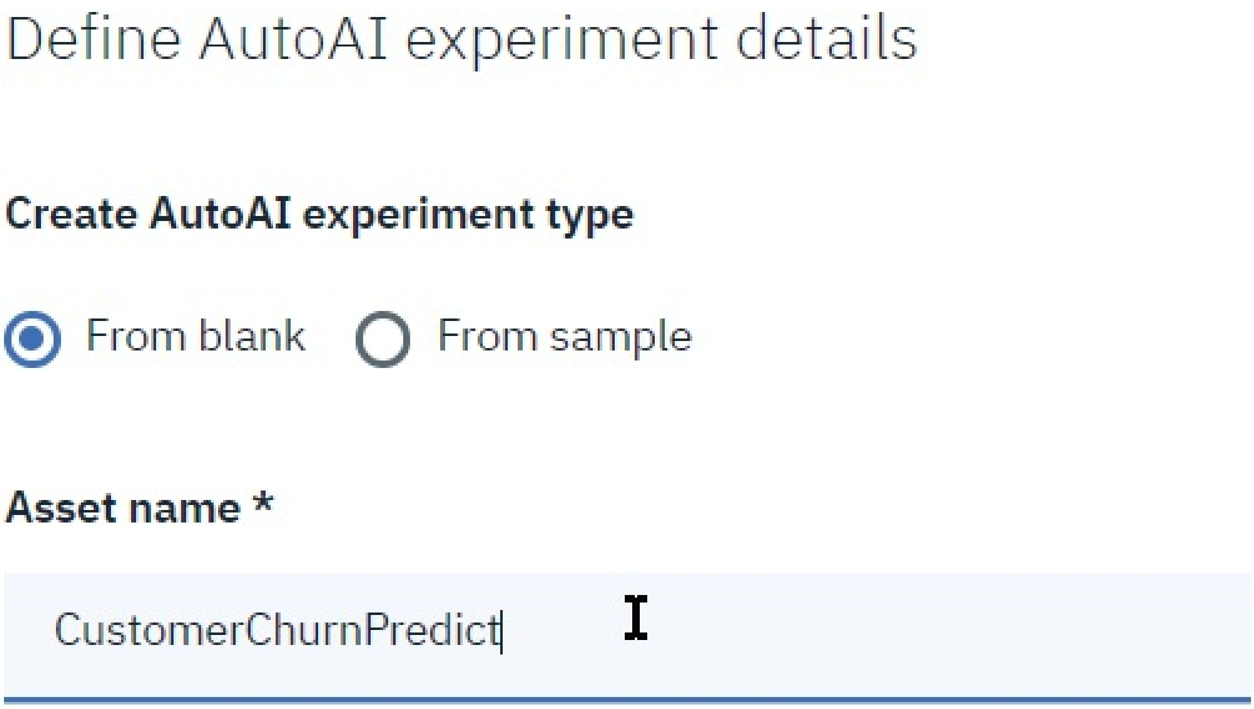
1. Use the `[(*) Add to Project]` button to bring up the artefact creation panel.
2. Create an Auto AI Experiment through the button



AutoAI experiment

Fully automated approach to building a classification or regression model.

3. On the `Create an AutoAI experiment` page, enter a name for the model, `CustomerChurnPredict` for example



Define AutoAI experiment details

Create AutoAI experiment type

● From blank ○ From sample

Asset name *

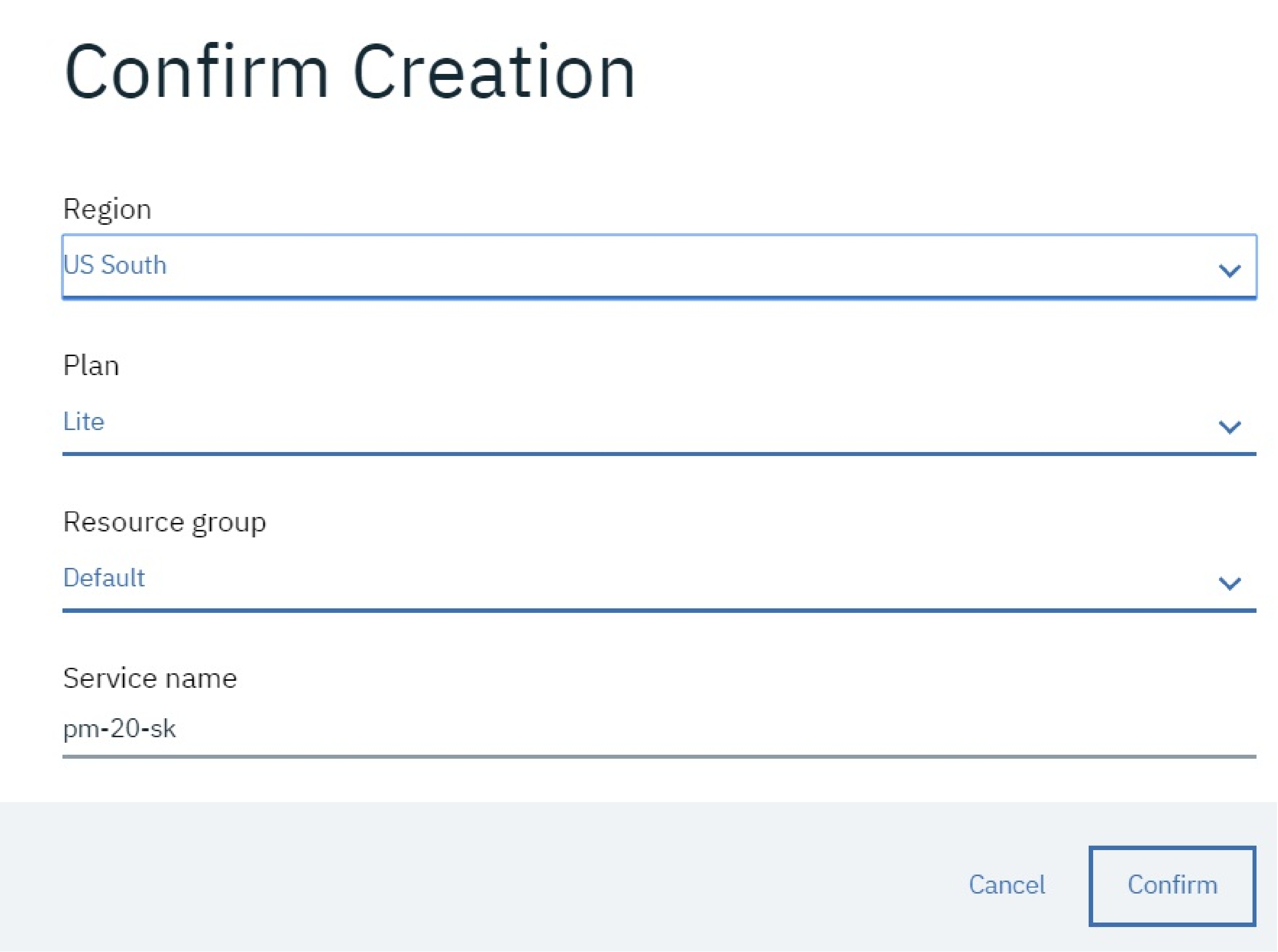
CustomerChurnPredict

4. An IBM Watson Machine Learning Service is required:
   i. Click on `Associate a Machine Learning service instance`

Associate a Machine Learning service instance

then click the reload button below to refresh th

   ii. Create a new Lite/Free plan instance:



Confirm Creation

Region
US South

Plan
Lite

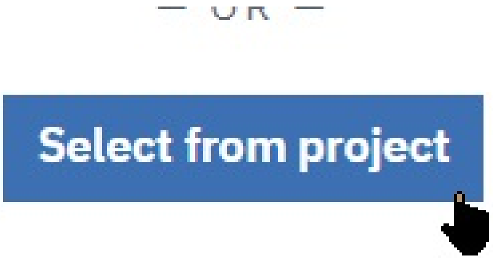
Resource group
Default

Service name
pm-20-sk

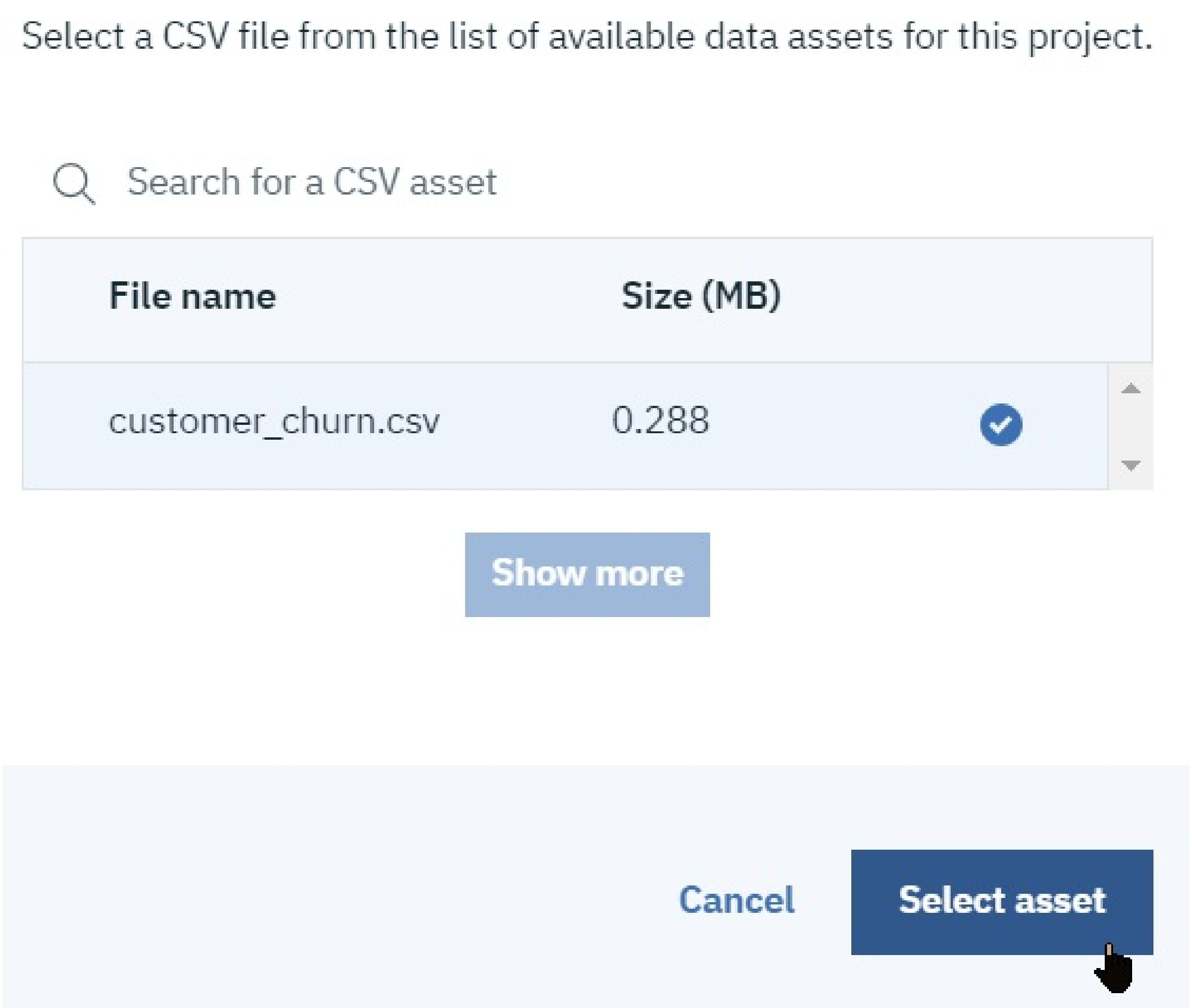
iii. Once created, switch back to the AutoAI creating tab and click `[Reload]`

to select your newly created instance.

5. Click `[Create]` button at the bottom right
6. On the "Add data source" page, you can select the data asset to use to create your model.
   - Since we have uploaded the file as a data asset already, we'll use



   - Highlight `customer_churn.csv` and `[Select asset]` button

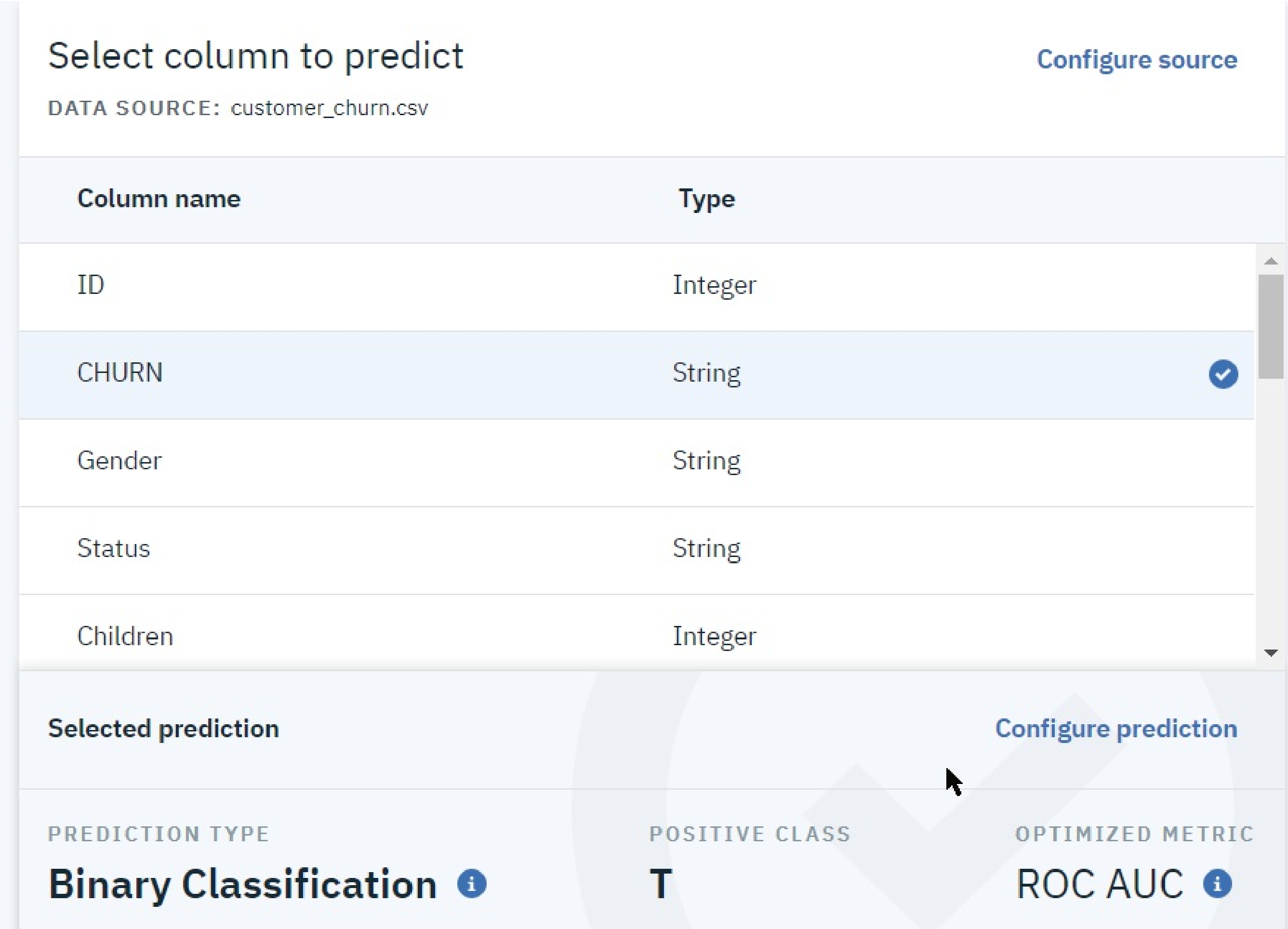


Select a CSV file from the list of available data assets for this project.

File name	Size (MB)	
customer_churn.csv	0.288	✓

- You could have used the browse button to upload `customer_churn.csv` if it had not been done earlier in this lab.

7. We will now configure the AutoAI input to drive the machine-learning predictive model construction. From the Select column to predict list, select CHURN. This is the column that contains the historical observations and thus the outcome to predict



Select column to predict

Configure source

DATA SOURCE: customer_churn.csv

Column name	Type	
ID	Integer	
CHURN	String	✓
Gender	String	
Status	String	
Children	Integer	

Selected prediction

Configure prediction

PREDICTION TYPE
Binary Classification

POSITIVE CLASS
T

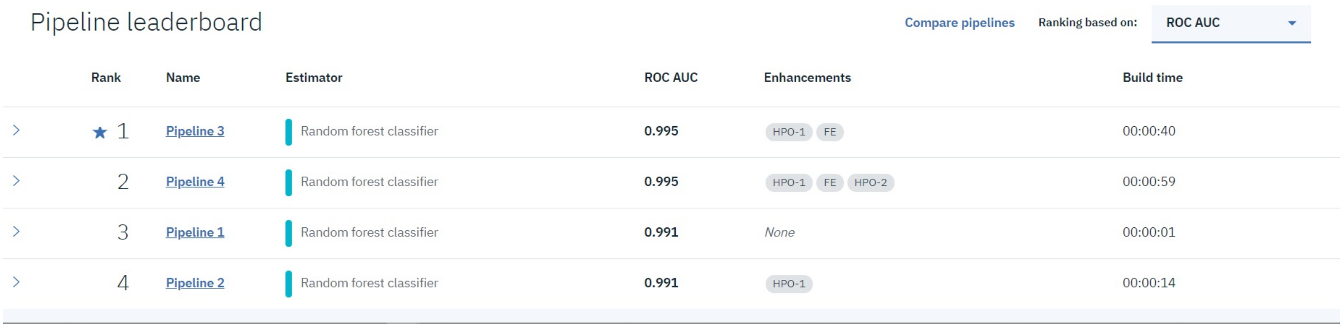
OPTIMIZED METRIC
ROC AUC

The columns contain the attributes on which the machine learning model will base predictions. All columns (features) that are not part of the prediction will be possible canditates for the prediction. We will see later on that AutoAI can help determine which ones are more pertinent than others.

8. As you can see, the AutoAI model builder selects Binary Classification by default as the type of model to build, because the `CHURN` column has been introspected and found to contain only two values, `T` and `F`. The model also selects `ROC AUC` as the metric for model evaluation. You could change those defaults under the `[Experiment settings]` button.
9. Now click on the `[Run experiment]` button

.

10. The canditate models will display in the pipeline leaderboard as they are evaluated



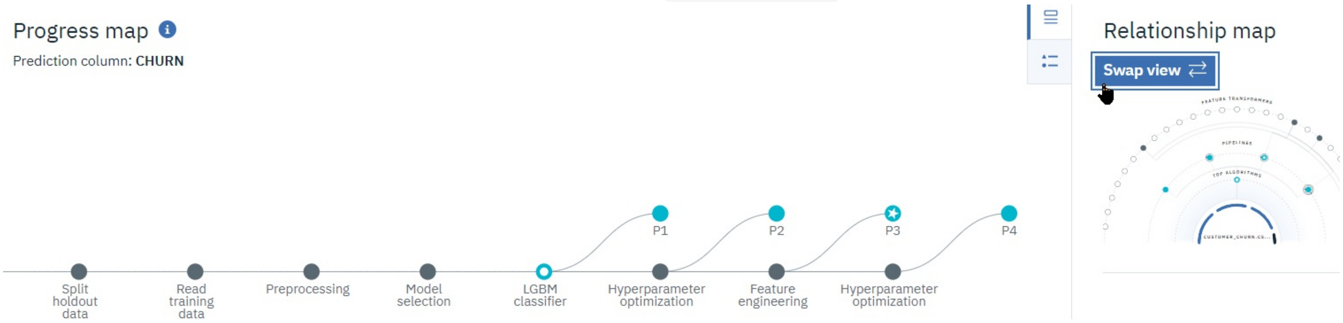
Pipeline leaderboard

Compare pipelines   Ranking based on: ROC AUC

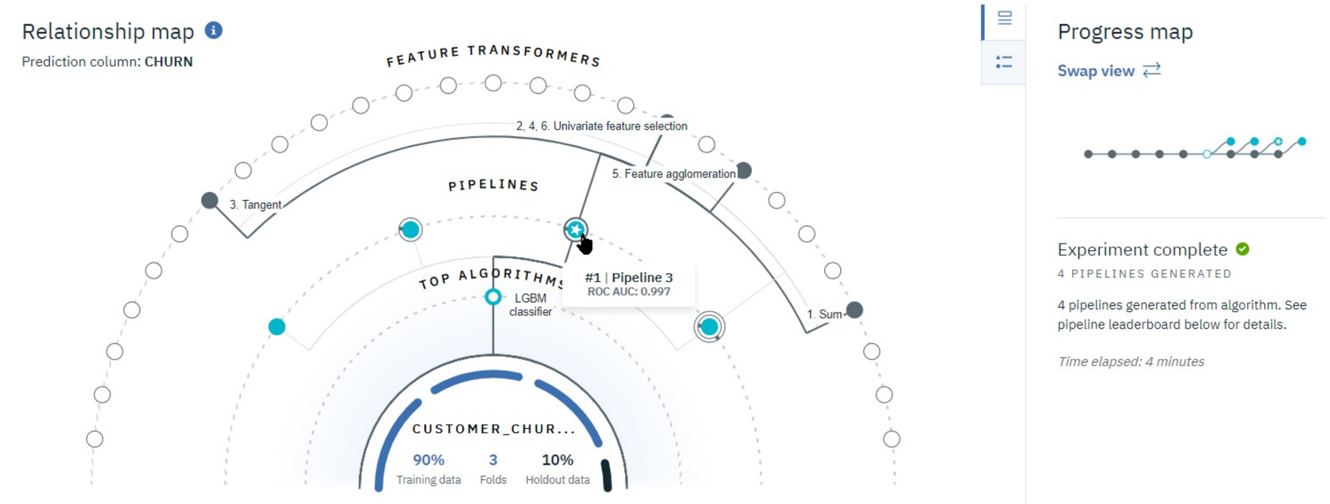
Rank	Name	Estimator	ROC AUC	Enhancements	Build time
★ 1	Pipeline 3	Random forest classifier	0.995	HPO-1 FE	00:00:40
2	Pipeline 4	Random forest classifier	0.995	HPO-1 FE HPO-2	00:00:59
3	Pipeline 1	Random forest classifier	0.991	*None*	00:00:01
4	Pipeline 2	Random forest classifier	0.991	HPO-1	00:00:14

, and ranked according to the selected metric (ROC AUC here by default). The ROC (Receiver Operating Characteristic) and PR (Precision Recall) Area Under Curve (AUC) are metrics used to evaluate the accuracy of a model's true positive and true negative predictions, evaluated on the test subset. The closer they are to 1.0, the better the sensitivity of the model.

11. This will trigger AutoAI's evaluation of the possible algorithms implementations and their configurations in order to select the best fiting one. The model evaluation and selection process is displayed as it executes (you may want to use the `[Swap View]` button)



12. Once completed, the relationship map shows the various paths that have been explored, highlighting the most accurate one as Top performer, hovering over the corresponding icon will reveal the selected algorithm (*LGBM CLassifier* here), and which *Feature Engineering* has been applied



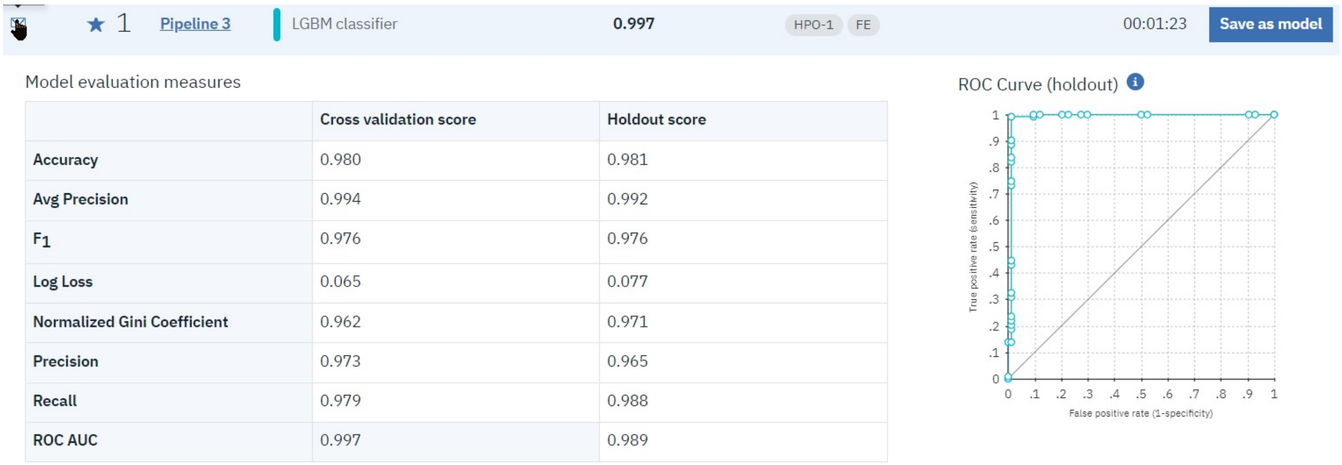
Experiment complete ✓

4 PIPELINES GENERATED

4 pipelines generated from algorithm. See pipeline leaderboard below for details.

Time elapsed: 4 minutes

13. We can get more details on a given canditate model by expanding it:



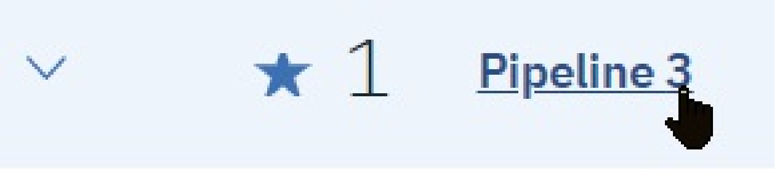
★ 1 Pipeline 3 | LGBM classifier 0.997 HPO-1 FE 00:01:23

Model evaluation measures

	Cross validation score	Holdout score
Accuracy	0.980	0.981
Avg Precision	0.994	0.992
F_1	0.976	0.976
Log Loss	0.065	0.077
Normalized Gini Coefficient	0.962	0.971
Precision	0.973	0.965
Recall	0.979	0.988
ROC AUC	0.997	0.989

. You will notice the AUC ROC plotout, and the metric computations on the cross-validation data set, which has been used for training, as well as the holdout, which has never been seen by the training and gives an idea of the atcual accuracy of the model.

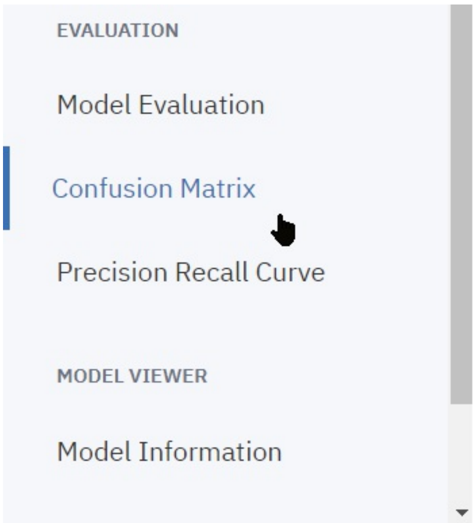
14. Looking deeper into the selected model yields more insights on your data set. Select the top pipeline

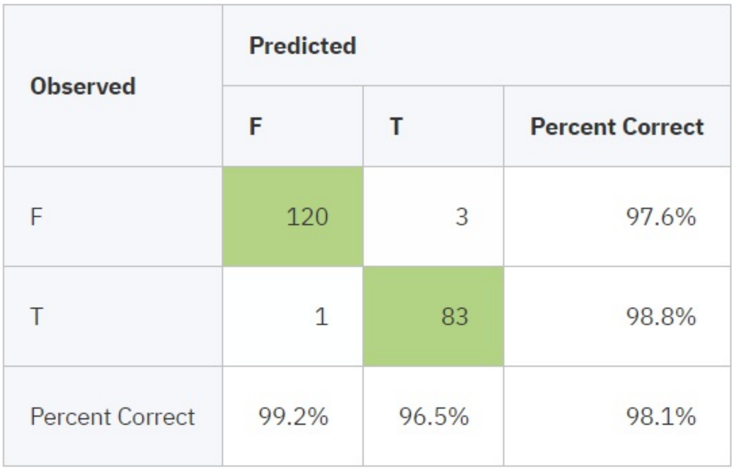


.

15. This opens the model details. Several tabs are of particular interest:

- The *Confusion Matrix*

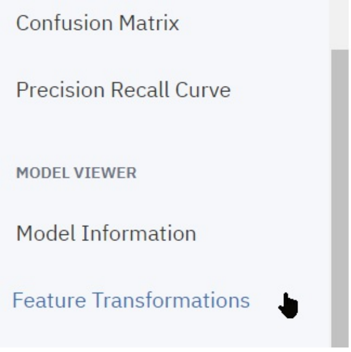


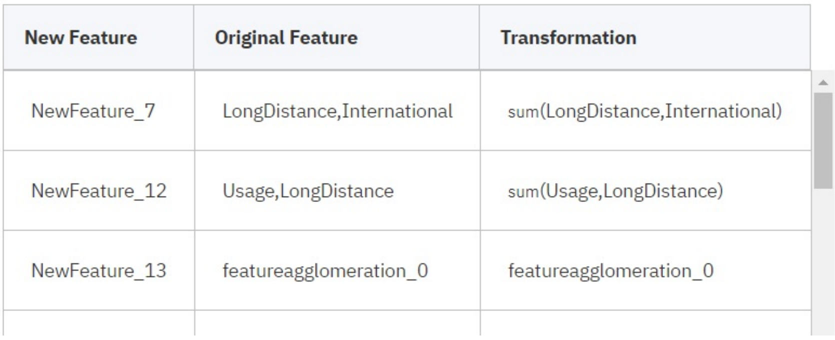


Observed	Predicted		
	F	T	Percent Correct
F	120	3	97.6%
T	1	83	98.8%
Percent Correct	99.2%	96.5%	98.1%

  shows more details on the model evaluation on the Holdout set. Here we can see that out of the 10% of rows set aside for the holdout set, in this case 206 rows, only 4 have been predicted wrong, for an overall accuracy of 98.1%.

- The *Features Transformations*

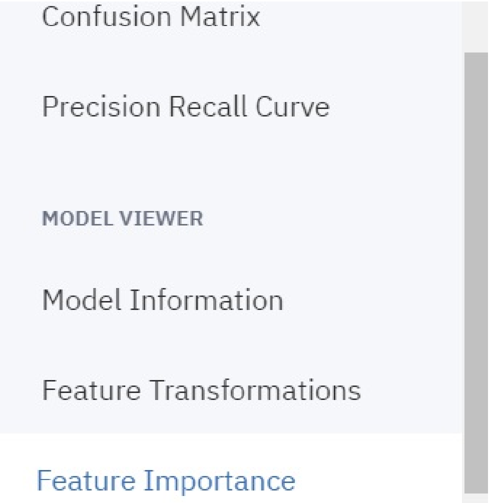


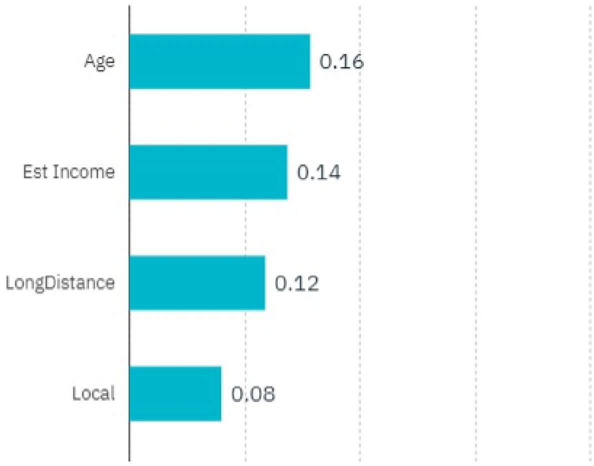


New Feature	Original Feature	Transformation
NewFeature_7	LongDistance,International	sum(LongDistance,International)
NewFeature_12	Usage,LongDistance	sum(Usage,LongDistance)
NewFeature_13	featureagglomeration_0	featureagglomeration_0

  tab shows which features AutoAI has generated, ranked by order of importance.
  It is interesting to find out that the sum of `LongDistance` and `International` calls appears as a relevant combined feature.

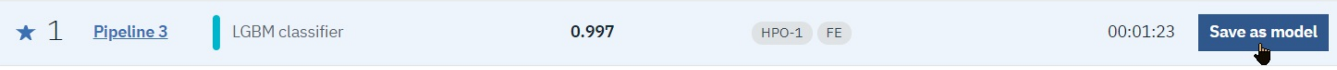
- The *Feature Importance*





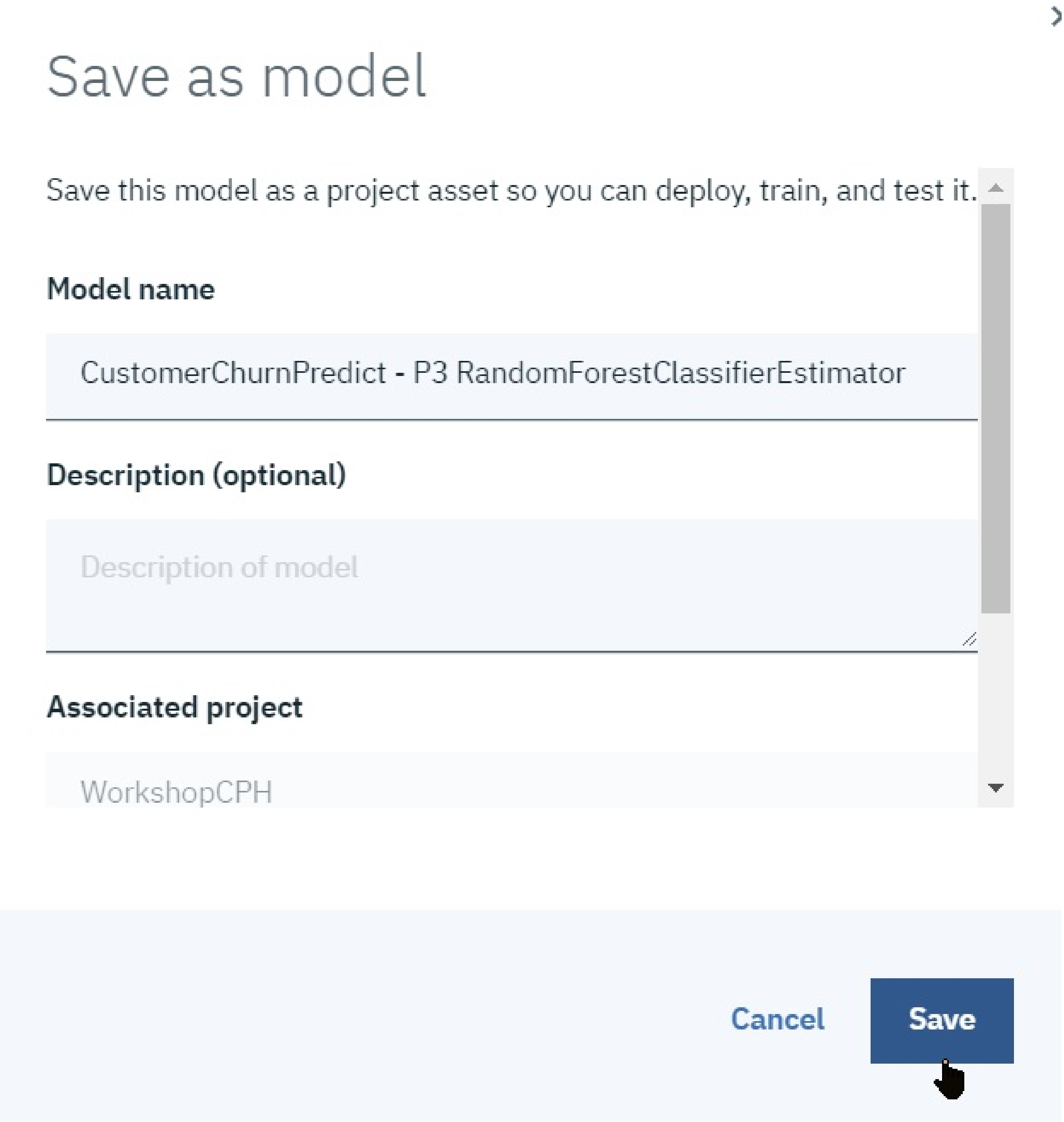
  shows which factors influence the most the `CHURN` target, in this case `Age` , `Est Income` , `LongDistance` account for 16%, 145 and 12% of the predictive power.

16. Lastly, we will select the best performing model (which usually has the best AUC ROC), and use the `[Save as model]` button to retain it



★ 1 Pipeline 3 | LGBM classifier | 0.997 | HPO-1 FE | 00:01:23 | Save as model

17. When you're prompted to confirm, click Save again



Save as model

Save this model as a project asset so you can deploy, train, and test it.

Model name

CustomerChurnPredict - P3 RandomForestClassifierEstimator

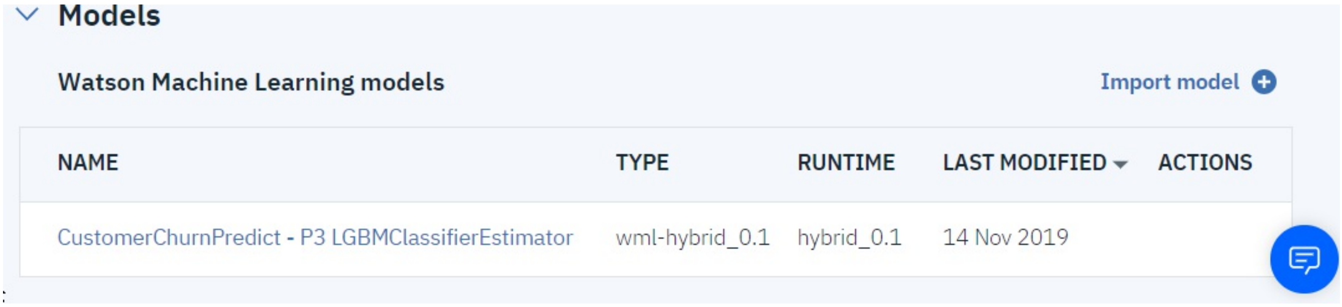
Description (optional)

Description of model

Associated project

WorkshopCPH

18. Switching back to your project's assets list, you will now find the saved trained model in the Models section:



Models

Watson Machine Learning models

NAME	TYPE	RUNTIME	LAST MODIFIED	ACTIONS
CustomerChurnPredict - P3 LGBMClassifierEstimator	wml-hybrid_0.1	hybrid_0.1	14 Nov 2019	

:

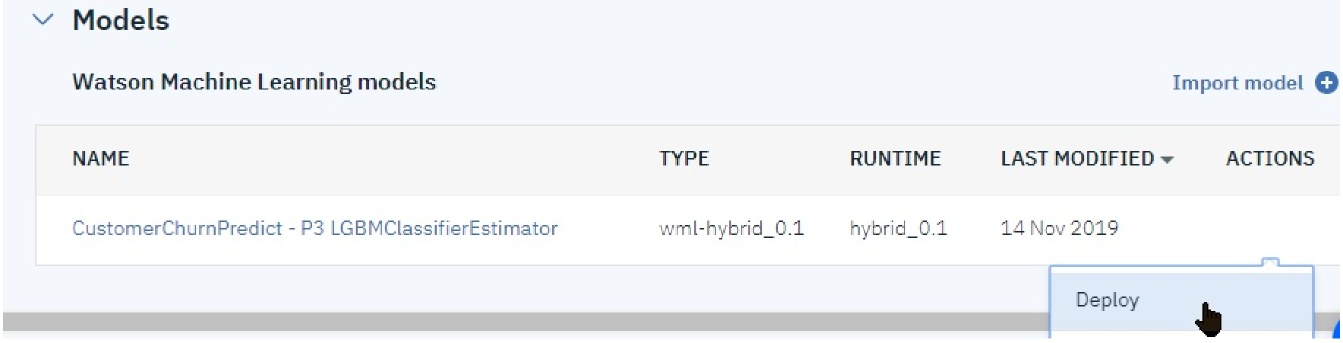
Using `View in project` link , you can be returned to the model's summary page, where your new model is shown.

> > You now have a trained model, next you will deploy the model to test on out-of-sample data.

[B] Deploy and test the trained model

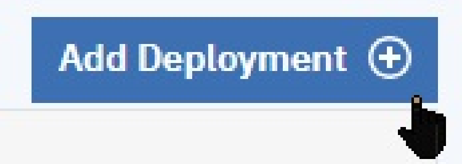
Before you can use your trained model to make predictions on new data, it must be made accessible from an Application Programming Interface (API).
This step is called Deployment, and it hinges between development and operations.

[B.1] Deploying the AutoAI predictive model

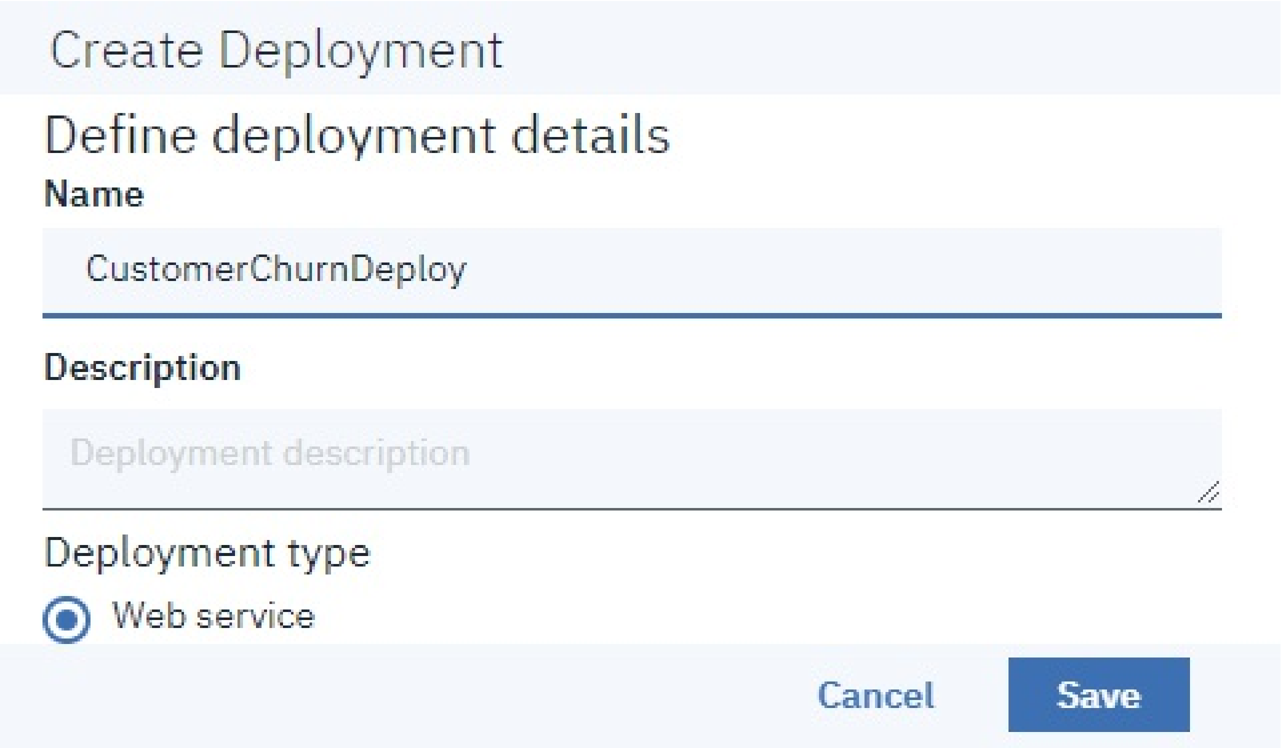
1. From the models list, locate your model and select the `Deploy` action

Models

Watson Machine Learning models

NAME	TYPE	RUNTIME	LAST MODIFIED	ACTIONS
CustomerChurnPredict - P3 LGBMClassifierEstimator	wml-hybrid_0.1	hybrid_0.1	14 Nov 2019	

2. You are taken to your model's *Deployments* tab. Click Add Deployment (+) link on the upper-right section of the pane 

3. On the Create Deployment page, give a name and a description to your deployment, e.g. `CustomerChurnDeploy`.
4. Clicking `[Save]` will deployed the model as a REST endpoint, as defined by Web service selection:



Create Deployment

Define deployment details

Name

CustomerChurnDeploy

Description

Deployment description

Deployment type

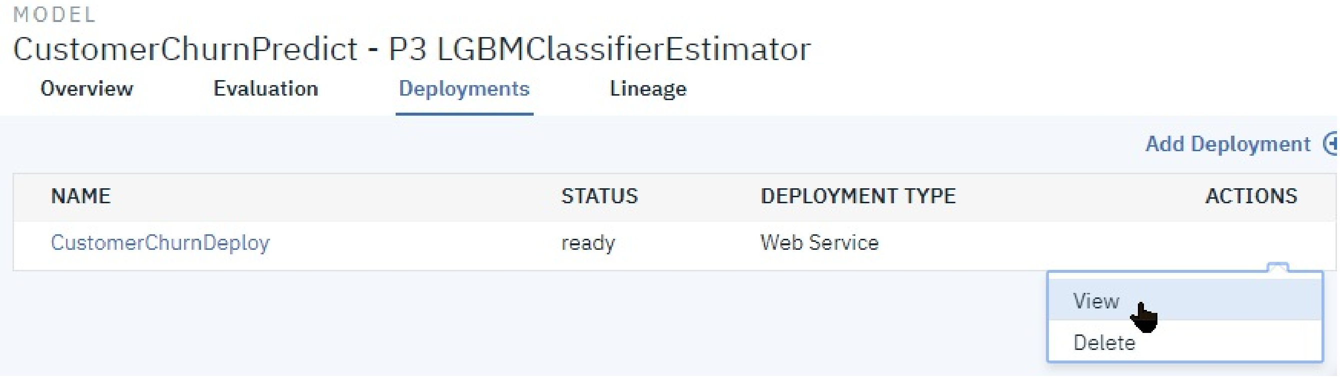
● Web service

Cancel

Save

.

5. When model deployment is complete, the STATUS turns to `ready` , and from the Actions menu, click View:



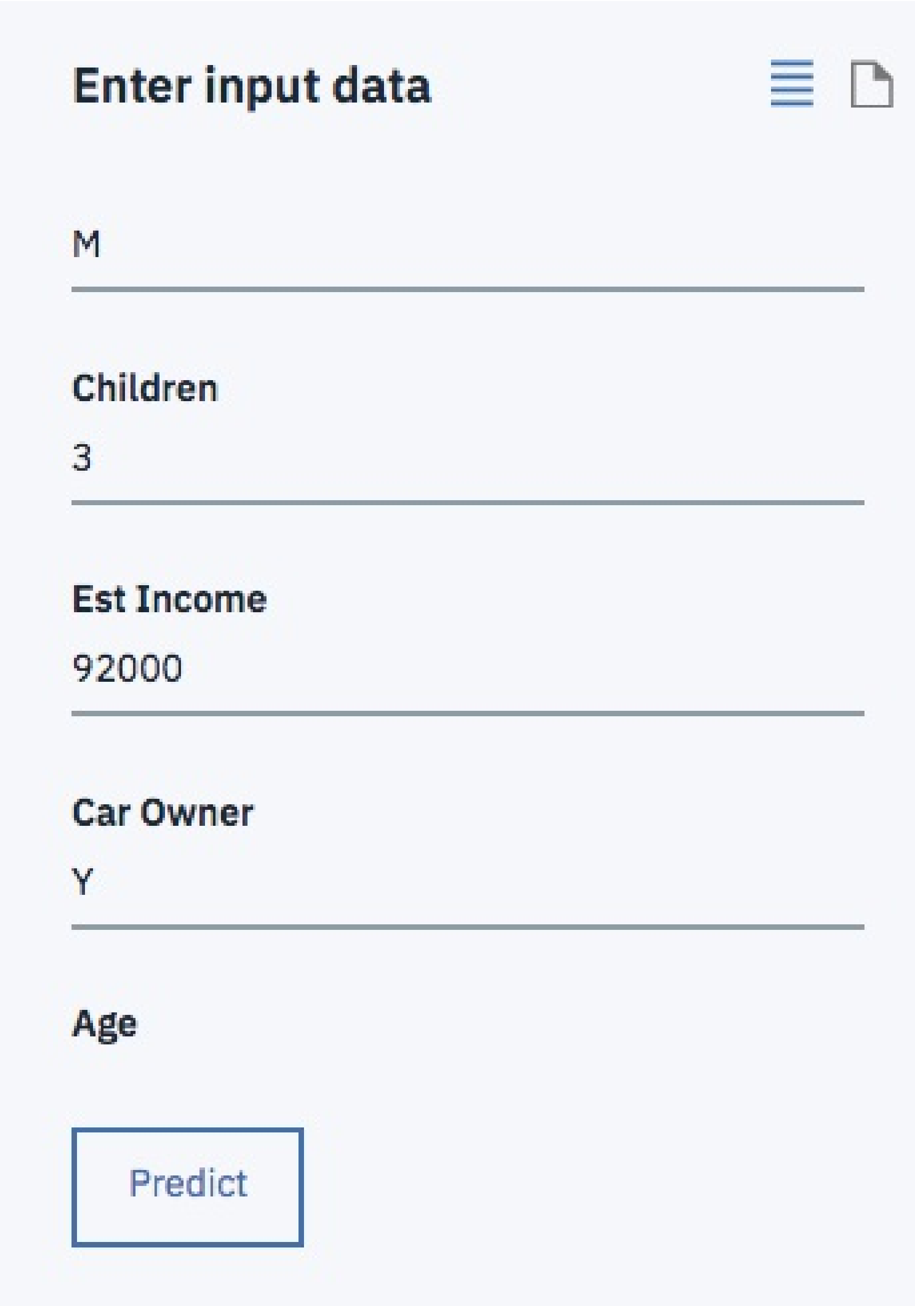
MODEL

CustomerChurnPredict - P3 LGBMClassifierEstimator

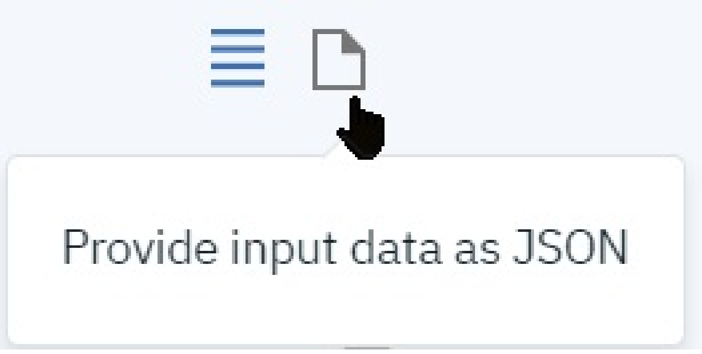
NAME	STATUS	DEPLOYMENT TYPE	ACTIONS
CustomerChurnDeploy	ready	Web Service	

If you are a developer, you may want to review the information in the Implementation tab.

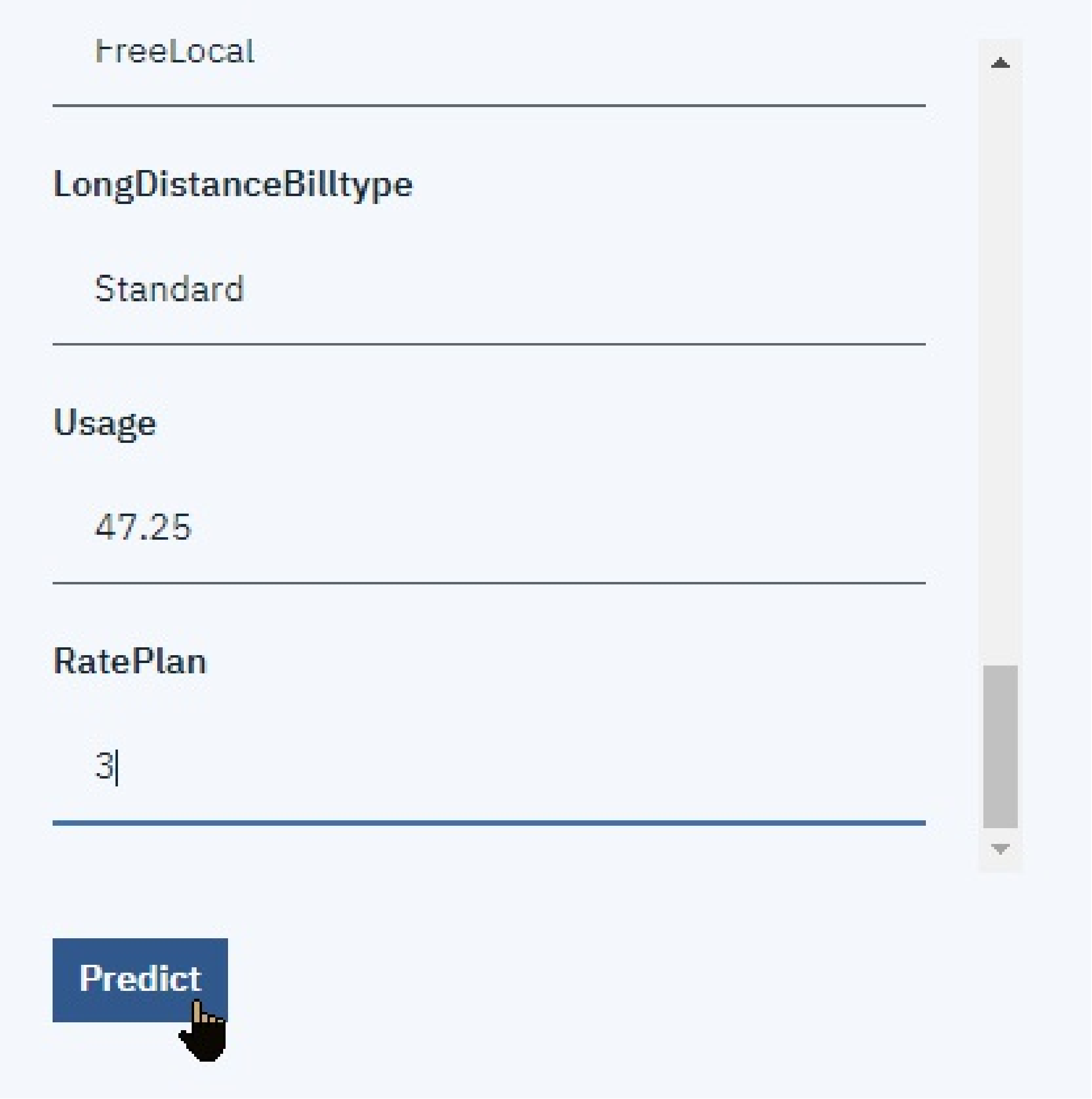
> The Code Snippets shows examples of how to use the REST endpoints from several environments, and can be passed on to you application developer to integrate the deployed model into a business application. We'll come back to it in the last section of this lab.
>
> 1. We are now able to the model prediction: go to Test tab.
> 2. Enter data in all the fields for a sample record from the data set.

> 
>
> Enter input data
>
> M
>
> Children
>
> 3
>
> Est Income
>
> 92000
>
> Car Owner
>
> Y
>
> Age
>
>
>
> For example, get data values from one line of the `new_customer_churn_data.csv` file, e.g:

```
8
F
M
0
19732.80
N
50.67
24.81
0
22.44
0
CC
FreeLocal
Standard
47.25
3
```

1. Note that alternatively, it is less tedious to switch to the raw JSON input using 

   Provide input data as JSON

   and copy the data from here and paste into the `Input JSON Payload data`: `JS {"input_data":[{"fields": ["ID", "Gender", "Status", "Children", "Est Income", "Car Owner","Age", "LongDistance", "International", "Local", "Dropped","Paymethod", "LocalBilltype", "LongDistanceBilltype","Usage", "RatePlan"],"values": [[6, "M", "M", 2, 29616, "N",49.42, 29.78, 0, 45.5, 0,"CH", "FreeLocal", "Standard",75.29, 2]]}]}`
2. To test the model and make a `CHURN` prediction on this data, click the `[Predict]` button: 

   FreeLocal

   LongDistanceBilltype

   Standard

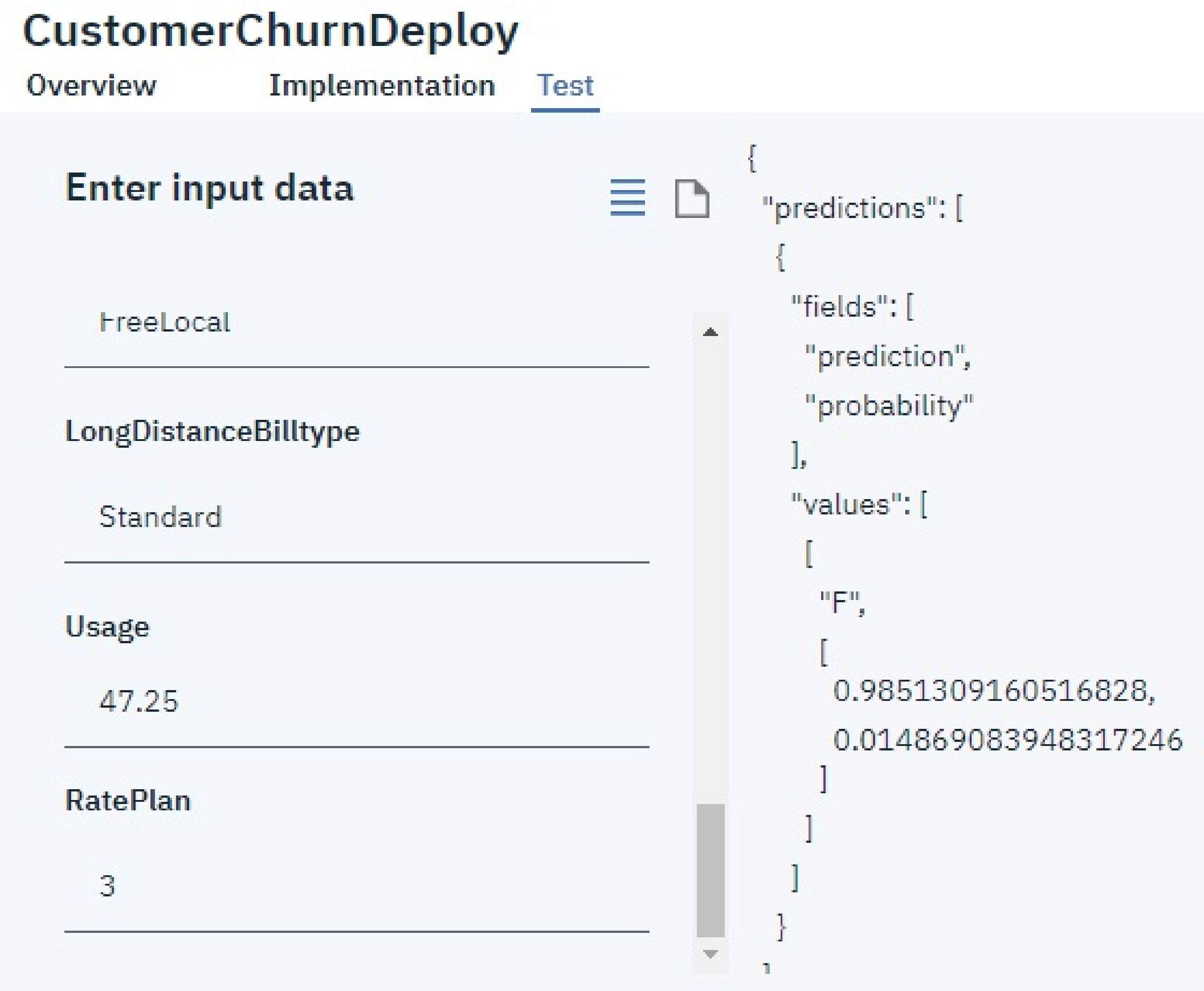
   Usage

   47.25

   RatePlan

   3

3. The system will invoke the Watson Machine Learning REST endpoint for the AutoAI model, you will get the resulting prediction buffer, looking like:



CustomerChurnDeploy

FreeLocal

LongDistanceBilltype

Standard

Usage

47.25

RatePlan

3

```
{
 "predictions": [
  {
   "fields": [
    "prediction",
    "probability"
   ],
   "values": [
    [
     "F",
     [
      0.9851309160516828,
      0.014869083948317246
     ]
    ]
   ]
  }
```

Here the prediction is that customer `CHURN` is `F` with a 98.5% probability.

4. Additionally, you can test several other records taken

[S] Optional Stretch Lab: run the model from a Python notebook

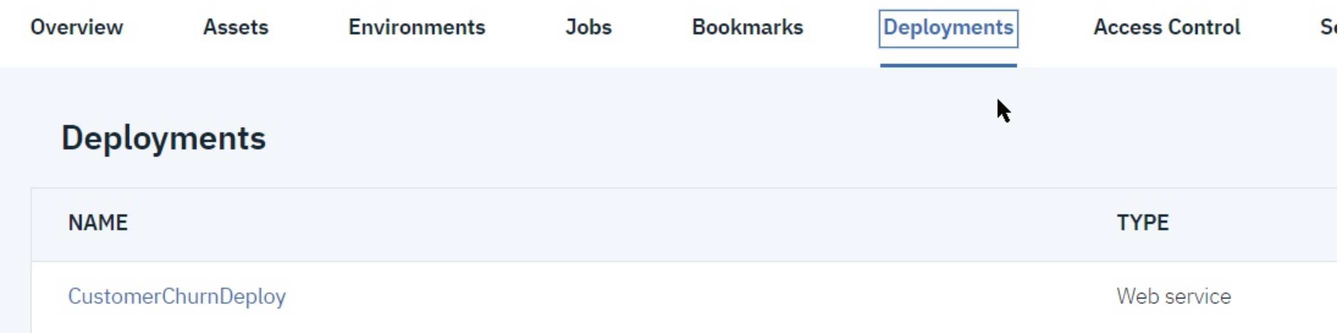
> Note: This section assumes that you have some proficiency in application development and will be comfortable dealing with some code!

The model just deployed can be invoked from any development environment that supports invoking REST endpoint, in the world of Data Science, the first intent will often be a Jupyter notebook coded in the Python language.

This is implemented in the `Lab-Stretch-RunModelFromNotebook.ipynb` notebook. This can be a first step towards integrating a ML model into an application.

Before creating the notebook, you will need to take note of the WML scoring REST endpoint for your model, and the Watson Machine Learning Service credentials:
**** [S.1] REST Scoring endpoint

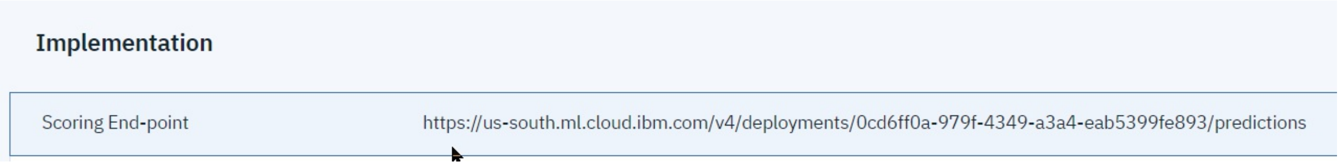
1. Select the Deployments tab in your project



Deployments

NAME	TYPE
CustomerChurnDeploy	Web service

2. Select the `CustomerChurnDeploy` created earlier, and switch to the
3. Select the `Implementation` tab, and copy the value of the `Scoring End-point` :



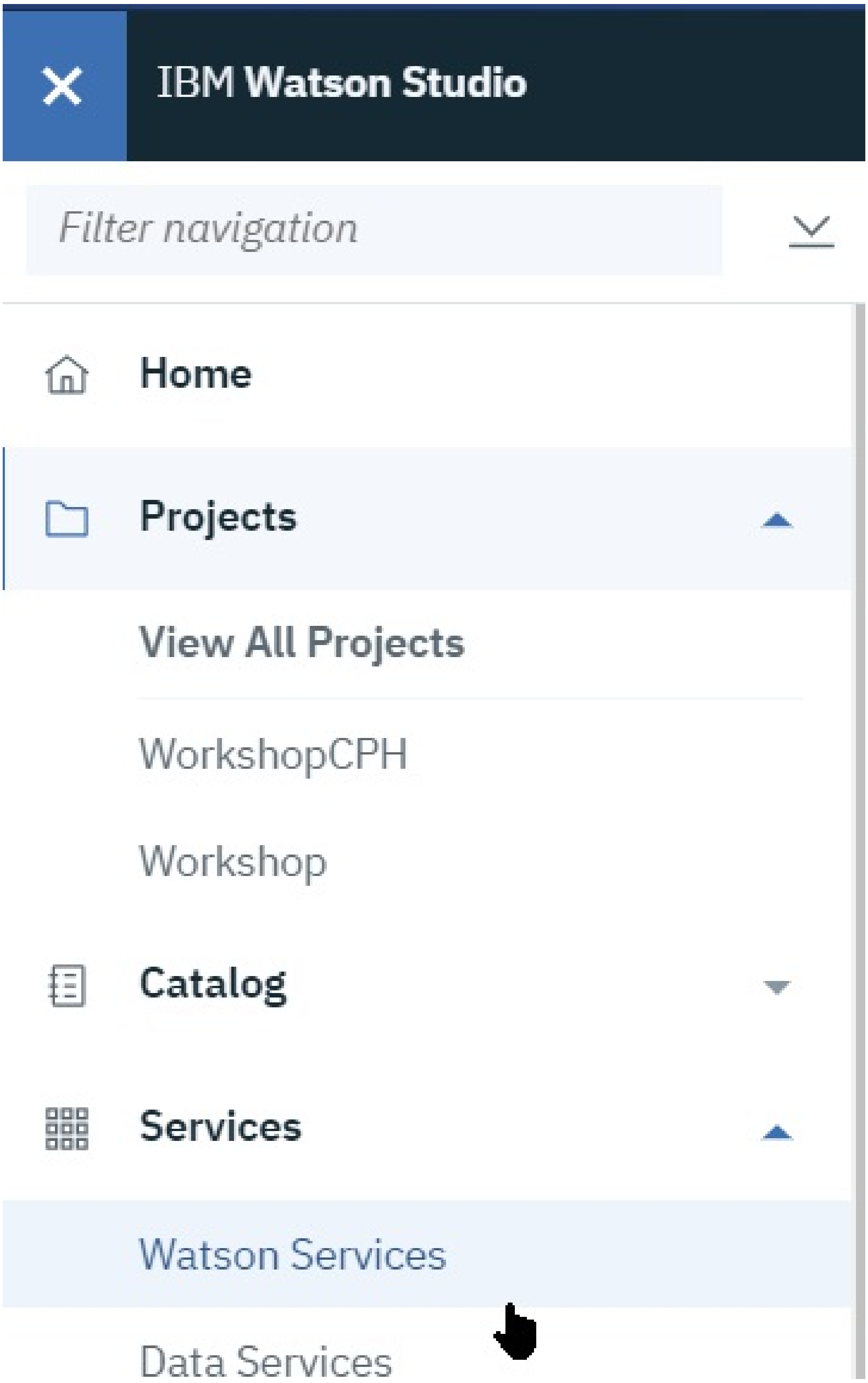
Implementation

Scoring End-point https://us-south.ml.cloud.ibm.com/v4/deployments/0cd6ff0a-979f-4349-a3a4-eab5399fe893/predictions

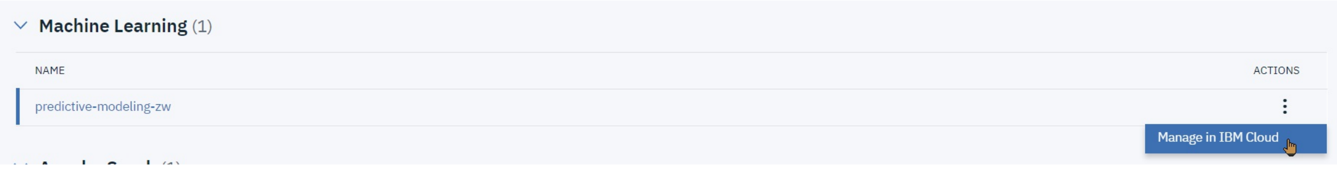
4. Paste this value in e.g. a notepad file on your laptop.

**** [S.2] Watson Machine Learning credentials

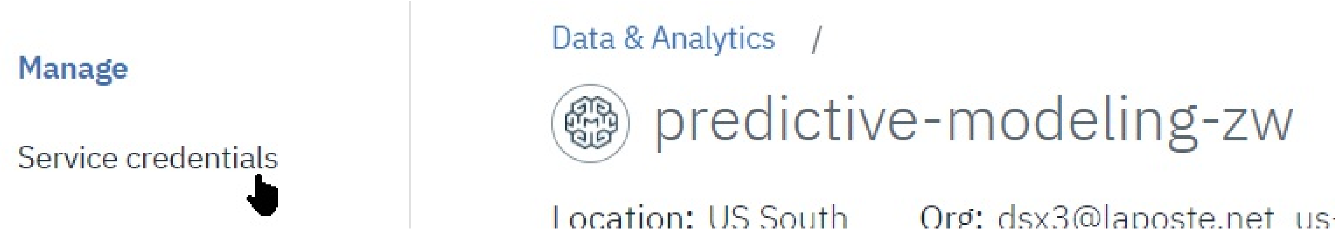
1. From the Hamburger menu (top left), select the `Services` menu and then the `Watson Services`



2. In the `Machine Learning` section, locate your service, and select `Manage in IBM Cloud` from its menu



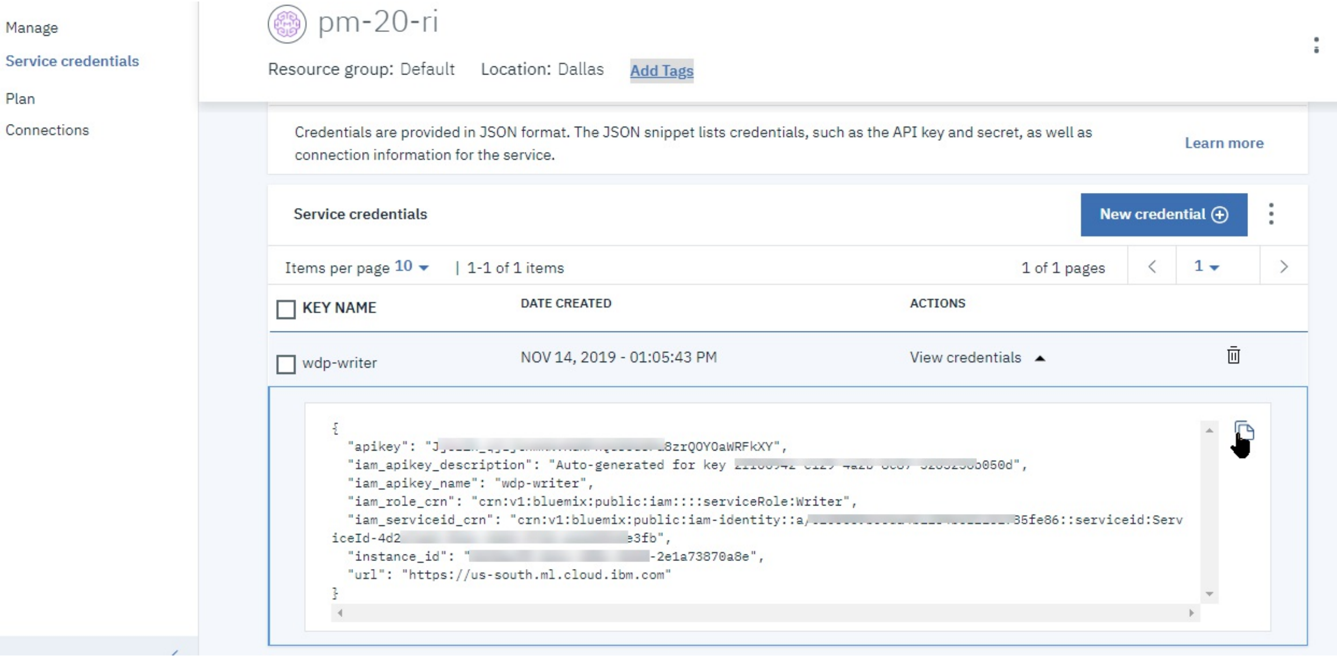
3. Select the `Service Credentials` tab



predictive-modeling-zw

Location: US South   Org: dsx3@laposte.net us-

4. Expand `View Credential`, and use the copy button (top right), and then paste them to a text file on your computer (using e.g. `notepad` on windows)



pm-20-ri

Resource group: Default   Location: Dallas   Add Tags

Credentials are provided in JSON format. The JSON snippet lists credentials, such as the API key and secret, as well as connection information for the service.   Learn more

Service credentials   New credential ⊕

Items per page 10 ▾ | 1-1 of 1 items   1 of 1 pages   ‹ 1 ▾ ›

□ KEY NAME	DATE CREATED	ACTIONS
□ wdp-writer	NOV 14, 2019 - 01:05:43 PM	View credentials ▲

```
{
  "apikey": "J[redacted]8zrQOYOaWRFkXY",
  "iam_apikey_description": "Auto-generated for key [redacted]050d",
  "iam_apikey_name": "wdp-writer",
  "iam_role_crn": "crn:v1:bluemix:public:iam::::serviceRole:Writer",
  "iam_serviceid_crn": "crn:v1:bluemix:public:iam-identity::a/[redacted]35fe86::serviceid:Serv
iceId-4d2[redacted]e3fb",
  "instance_id": "[redacted]-2e1a73870a8e",
  "url": "https://us-south.ml.cloud.ibm.com"
}
```

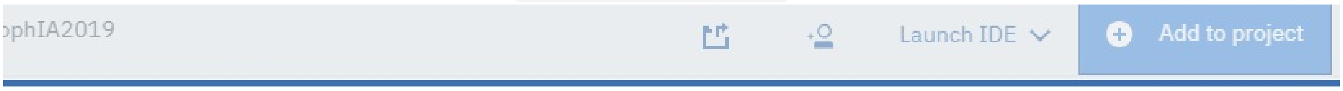
.

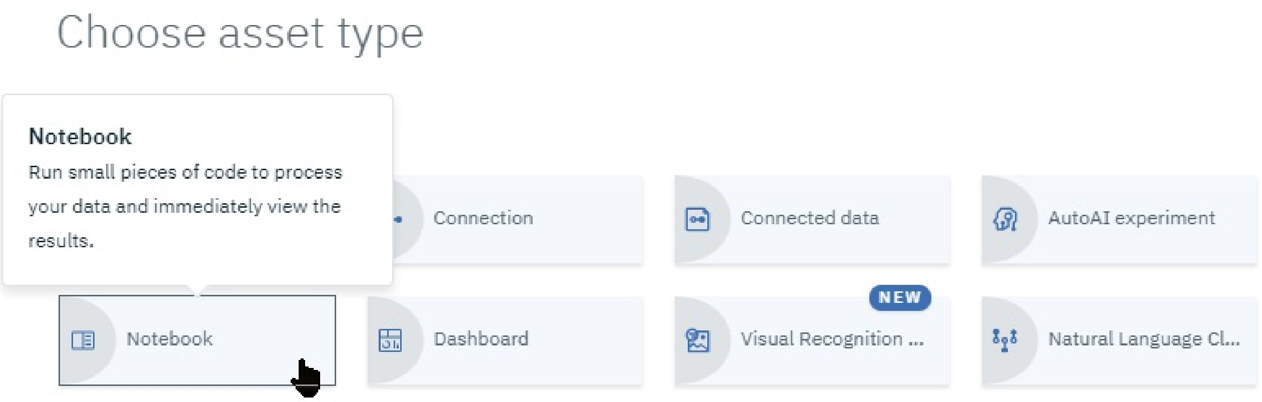
We will use these values in a python notebook shortly.

[S.3] Creating the notebook

Now you can switch back to Watson Studio and add a notebook from file:

1. From your project's Asset tab, use the `(+) Add to project` button to add a notebook

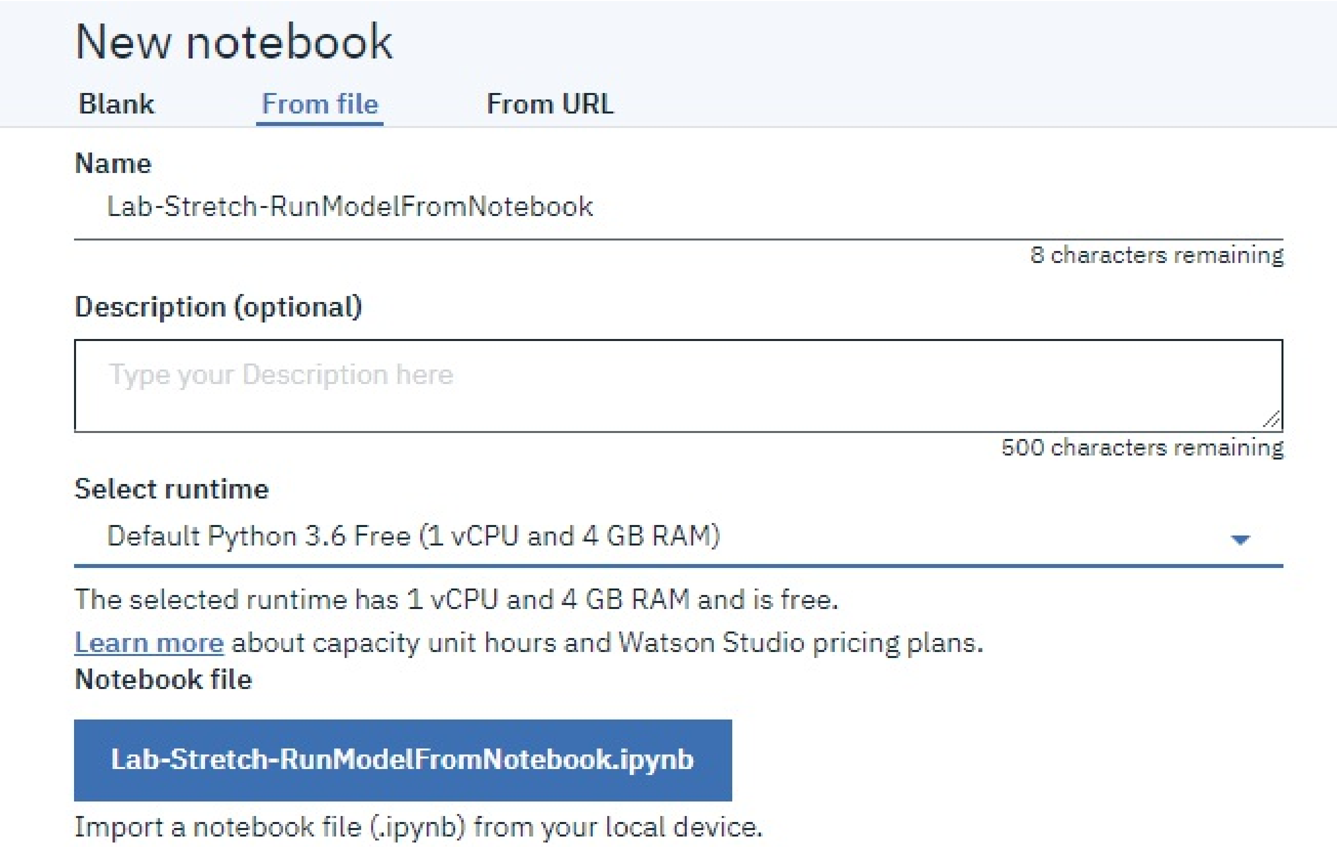




Choose asset type

Notebook
Run small pieces of code to process your data and immediately view the results.

2. Use the `From file` tab and `[Choose file]` button to load the `Lab-Stretch-ScoreModelFromNotebook.ipynb` notebook file
3. Select a `Default Python 3.6 Free` environment from the



New notebook

Name

Lab-Stretch-RunModelFromNotebook

8 characters remaining

Description (optional)

Type your Description here

500 characters remaining

Select runtime

Default Python 3.6 Free (1 vCPU and 4 GB RAM)

The selected runtime has 1 vCPU and 4 GB RAM and is free.

Learn more about capacity unit hours and Watson Studio pricing plans.

Notebook file

Lab-Stretch-RunModelFromNotebook.ipynb

Import a notebook file (.ipynb) from your local device.

4. Click `[Create Notebook]`, and follow the instructions within the notebook.

[S.4] Executing the notebook

Instructions are within the notebook itself, in comment cells.

In essence, you will execute the notebook's code cells using the `[(>) Run]` button.
Minimal changes will be required to specify the proper REST endpoint URL and WML credentials.

Conclusion

You completed the lab for Training and Deploying a model using the IBM Watson Studio's *AutoAI* model builder.