Tutorial: Create an AI/ML model with Azure Automated ML

In this tutorial, you learn how to create a powerful classification model without writing a single line of code using automated machine learning in the Azure Machine Learning studio. This classification model predicts if a customer will pay his/her account on time (i.e. before or on the invoice due date). Optional – you could also run a regression model and predict the actual number of days a customer will pay.

In this scenario, we have an SAP Accounts Receivable file and external creditor data that resides in an Azure Data Lake. We will work on the consolidated view and apply Azure AutoML on the data.

With automated machine learning, you can automate away time intensive tasks. Automated machine learning rapidly iterates over many combinations of algorithms and hyperparameters to help you find the best model based on a success metric of your choosing.

In this tutorial, you will go through the following tasks in order to complete the exercise end to end:

a) Create an Azure Machine Learning workspace in the Resource Group.
b) Launch Azure Machine Learning Workspace
c) Run an Automated Machine Learning experiment and leverage the SAP accounts receivable and creditor file.
d) View experiment details.
e) Deploy the model so that you have a REST API end point.

Prerequisites

- An Azure subscription. If you don't have an Azure subscription leverage the Azure Voucher to create a subscription.
- Download the **WA_Fn-UseC_-Accounts-ReceivableCreditScore.csv** data file on your local machine. The "**LateYorN**" column indicates if the customer was late or not with the payment. For our classification, this column will later be identified as the target column for predictions in this tutorial. Optional lab – we can also predict the "**DaysToSettle**" with a regression model that will predict how many days it took for a customer to pay.

Sample of the **WA_Fn-UseC_-Accounts-ReceivableCreditScore.csv** data file:

countryCode	customerID	PaperlessDate	invoiceNumber	InvoiceDate	DueDate	InvoiceAmount	Disputed	SettledDate	PaperlessBill	DaysToSettle	DaysLate	LateYorN	CreditRating
391	0187-ERLSR	7/31/2013	1756742390	9/5/2012	10/5/2012	84.57	No	9/14/2012	Paper	9	0	N	A
391	0187-ERLSR	7/31/2013	4037644863	3/29/2012	4/28/2012	62.68	Yes	4/25/2012	Paper	27	0	N	A
391	0187-ERLSR	7/31/2013	4063317759	9/22/2012	10/22/2012	65.26	Yes	10/11/2012	Paper	19	0	N	A
391	0187-ERLSR	7/31/2013	4160638076	2/16/2013	3/18/2013	56.5	Yes	3/2/2013	Paper	14	0	N	A
391	0187-ERLSR	7/31/2013	4814212537	3/22/2013	4/21/2013	86.92	No	3/27/2013	Paper	5	0	N	A

File can be downloaded in the following location:

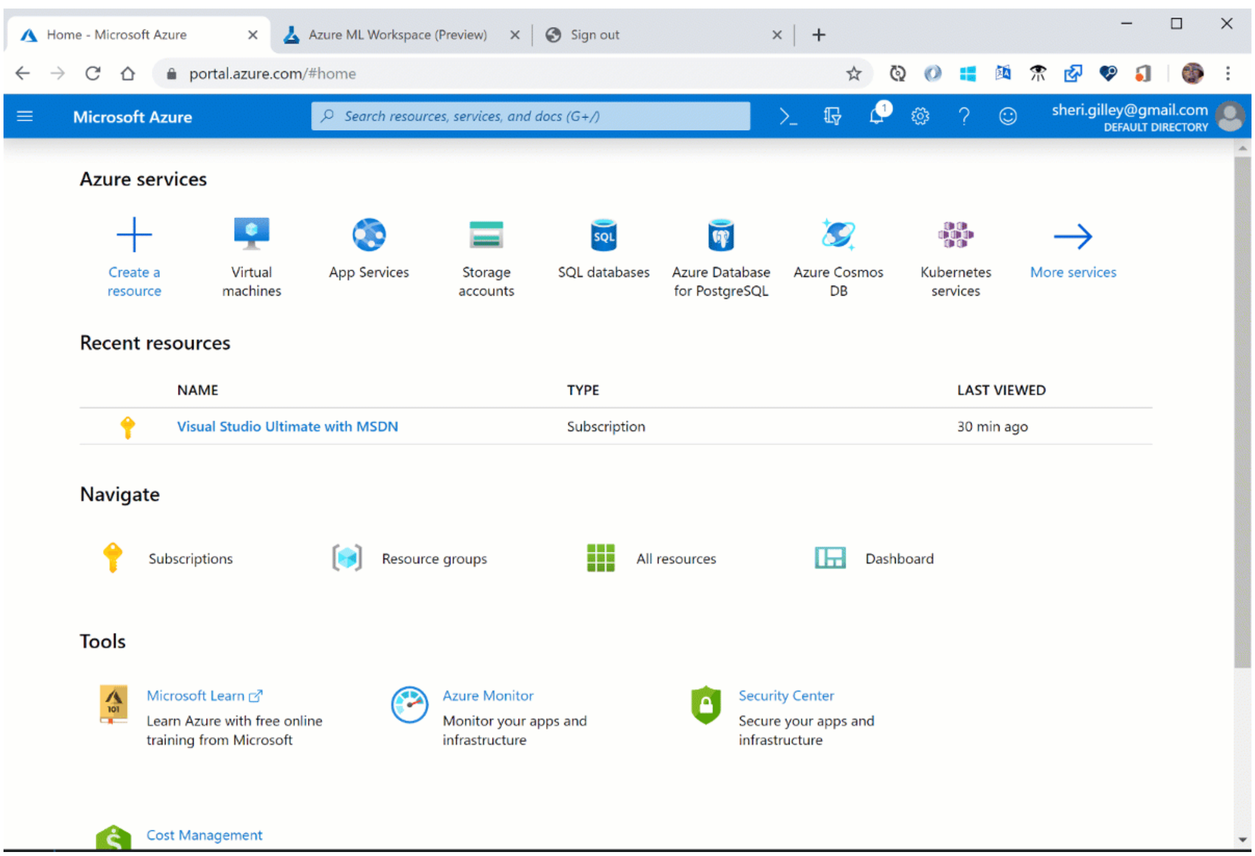
https://github.com/alpkayaMSFT/saponazureml/blob/main/WA_Fn-UseC_-Accounts-ReceivableCreditScore.csv

Create an Azure Machine Learning Workspace

An Azure Machine Learning workspace is a foundational resource in the cloud that you use to experiment, train, and deploy machine learning models. It ties your Azure subscription and resource group to an easily consumed object in the service.

There are many ways to create a workspace. In this tutorial, you create a workspace via the Azure portal, a web-based console for managing your Azure resources.

1. Sign in to the Azure portal by using the credentials for your Azure subscription.
2. In the upper-left corner of the Azure portal, select **+ Create a resource**.



3. Use the search bar to find **Machine Learning**.
4. Select **Machine Learning**.
5. In the **Machine Learning** pane, select **Create** to begin.
6. Provide the following information to configure your new workspace:

TABLE 1

Field	Description
Workspace name	Enter a unique name that identifies your workspace. In this example, we use **saponazuremachinelearningws**. Names must be unique across the resource group. Use a name that's easy to recall and to differentiate from workspaces created by others.
Subscription	Select the Azure subscription that you want to use.
Resource group	Use an existing resource group in your subscription, or enter a name to create a new resource group. A resource group holds related resources for an Azure solution. In this example, we use **saponazuremachinelearning**.
Location	Select the location closest to your users and the data resources to create your workspace.
Workspace edition	Select **Basic** as the workspace type for this tutorial. The workspace type determines the features to which you'll have access and pricing. Everything in this tutorial can be performed with either a Basic or Enterprise workspace.

7. After you're finished configuring the workspace, select **Review + Create**.

   Warning

   It can take several minutes to create your workspace in the cloud.

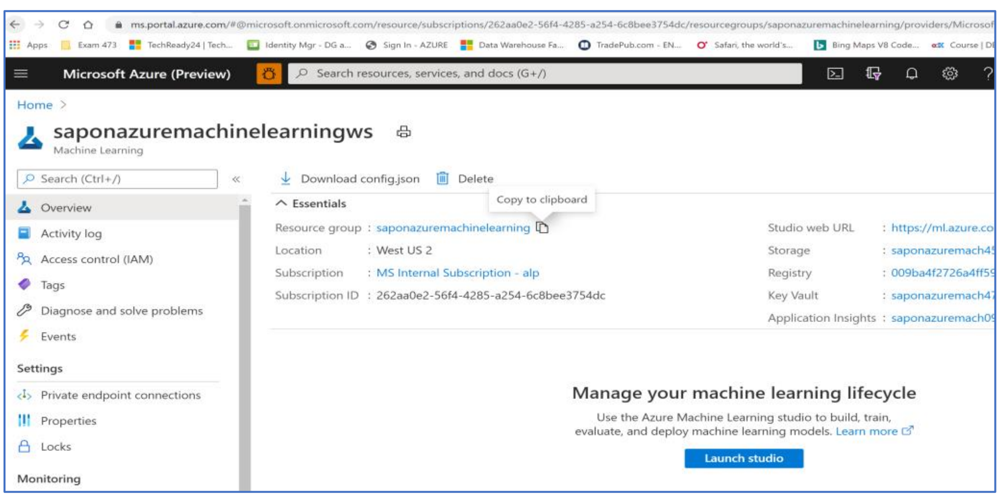
   When the process is finished, a deployment success message appears.

8. To view the new workspace, select **Go to resource**.

Get started in Azure Machine Learning studio

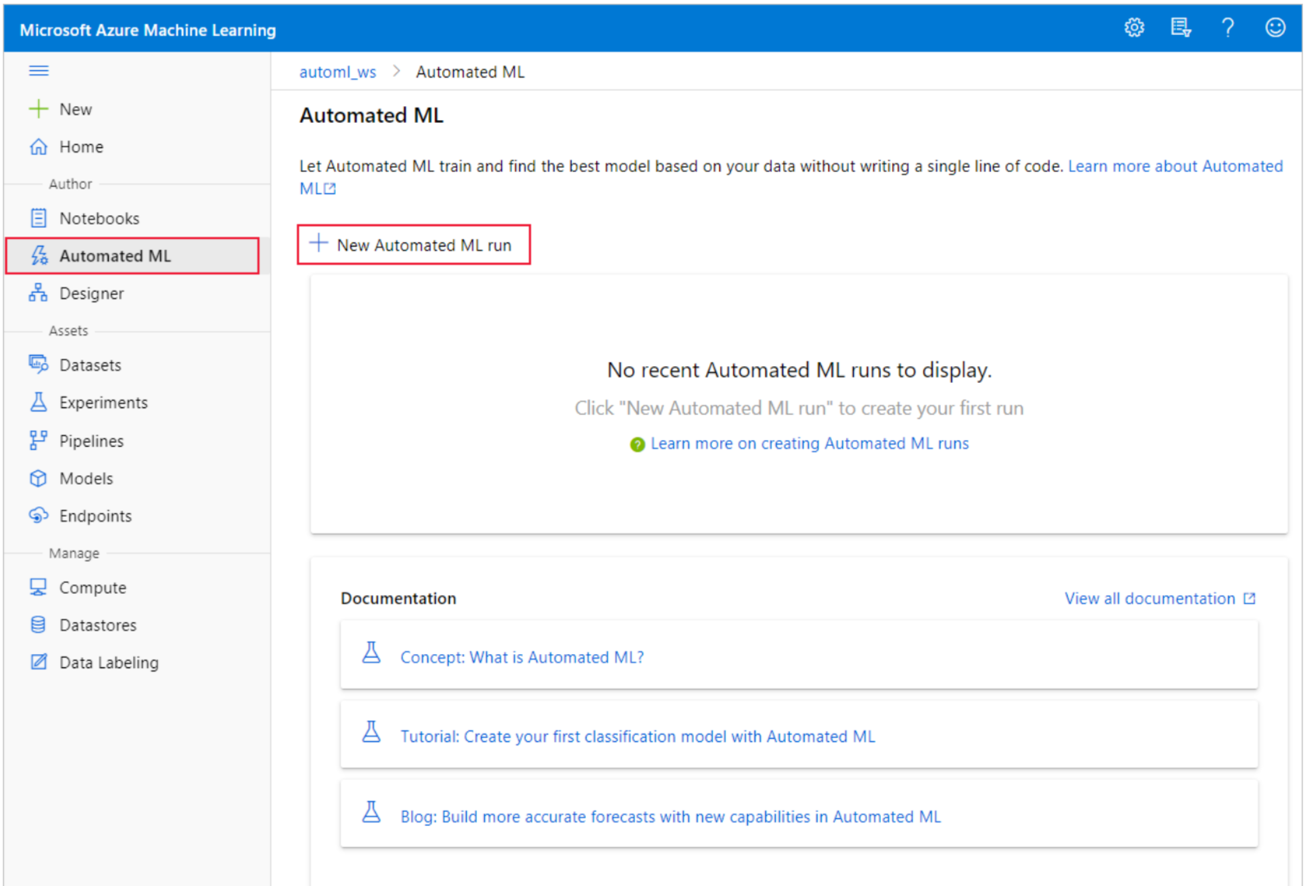
You complete the following experiment set-up and run steps via the Azure Machine Learning studio at https://ml.azure.com, a consolidated web interface that includes machine learning tools to perform data science scenarios for data science practitioners of all skill levels.

1. Sign in to Azure Machine Learning studio. Alternatively, you can click on launch Studio in the ML resource page you created earlier – see below:



2. Select your subscription and the ML workspace you created.
3. Select **Get started**.
4. In the left pane, select **Automated ML** under the **Author** section.

   Since this is your first automated ML experiment, you'll see an empty list and links to documentation.

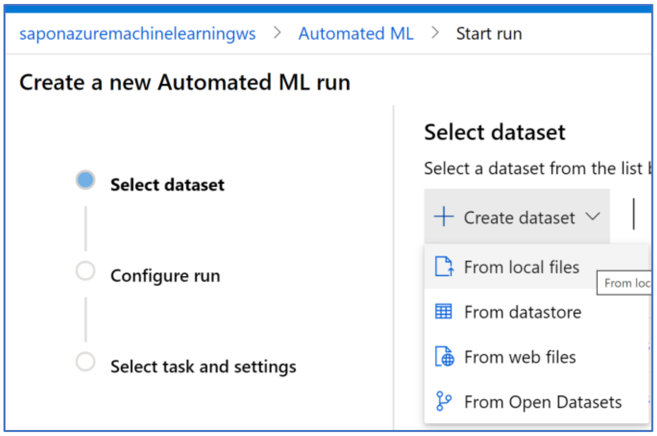


5. Select **+New automated ML run**.

Create and load dataset

Before you configure your experiment, upload your data file to your workspace in the form of an Azure Machine Learning dataset. Doing so, allows you to ensure that your data is formatted appropriately for your experiment.

1. Create a new dataset by selecting **From local files** from the **+Create dataset** drop-down.
   1. On the **Basic info** form, give your dataset a name and provide an optional description. The automated ML interface currently only supports TabularDatasets, so the dataset type should default to *Tabular*.



2. Select **Next** on the bottom left
3. On the **Datastore and file selection** form, select the default datastore that was automatically set up during your workspace creation, **workspaceblobstore (Azure Blob Storage)**. This is where you'll upload your data file to make it available to your workspace.
4. Select **Browse**.
5. Choose the **WA_Fn-UseC_-Accounts-ReceivableCreditScore.csv** file on your local computer. This is the file you downloaded as a prerequisite.
6. Give your dataset a unique name and provide an optional description.
7. Select **Next** on the bottom left, to upload it to the default container that was automatically set up during your workspace creation.

   When the upload is complete, the Settings and preview form is pre-populated based on the file type.

8. Verify that the **Settings and preview** form is populated as follows and select **Next**.

TABLE 2

Field	**Description**	**Value for tutorial**
File format	Defines the layout and type of data stored in a file.	Delimited
Delimiter	One or more characters for specifying the boundary between separate, independent regions in plain text or other data streams.	Comma
Encoding	Identifies what bit to character schema table to use to read your dataset.	UTF-8
Column headers	Indicates how the headers of the dataset, if any, will be treated.	First row is the header
Skip rows	Indicates how many, if any, rows are skipped in the dataset.	None

9. The **Schema** form allows for further configuration of your data for this experiment. For this example, we don't make any selections. Select **Next**.

10. On the **Confirm details** form, verify the information matches what was previously populated on the **Basic info, Datastore and file selection** and **Settings and preview** forms.
11. Select **Create** to complete the creation of your dataset.
12. Select your dataset once it appears in the list.
13. Select **Next**.

Configure run

After you load and configure your data, you can set up your experiment. This setup includes experiment design tasks such as, selecting the size of your compute environment and specifying what column you want to predict.

1. Select the **Create new** radio button.
2. Populate the **Configure Run** form as follows:

Enter this experiment name as you like we choose: sapaccountsreceiveablelateyesorno

1. Select "**LateYorN**" as the target column, what you want to predict. This column indicates whether the customer paid on time or not.
2. Select **+Create a new compute** and configure your compute target. A compute target is a local or cloud-based resource environment used to run your training script or host your service deployment. For this experiment, we use a cloud-based compute.
    1. Populate the **Virtual Machine** form to set up your compute.

TABLE 3

Field	Description	Value for tutorial
Virtual machine priority	Select what priority your experiment should have	Dedicated
Virtual machine type	Select the virtual machine type for your compute.	CPU (Central Processing Unit)
Virtual machine size	Select the virtual machine size for your compute. A list of recommended sizes is provided based on your data and experiment type.	Standard_DS12_V2

    2. Select **Next** to populate the **Configure settings form**.

TABLE 4

Field	Description	Value for tutorial
Compute name	A unique name that identifies your compute context.	saponazureml
Min / Max nodes	To profile data, you must specify 1 or more nodes.	Min nodes: 1 Max nodes: 6
Idle seconds before scale down	Idle time before the cluster is automatically scaled down to the minimum node count.	120 (default)

TABLE 4

Field	Description	Value for tutorial
Advanced settings	Settings to configure and authorize a virtual network for your experiment.	None

3. Select **Create** to create your compute target.

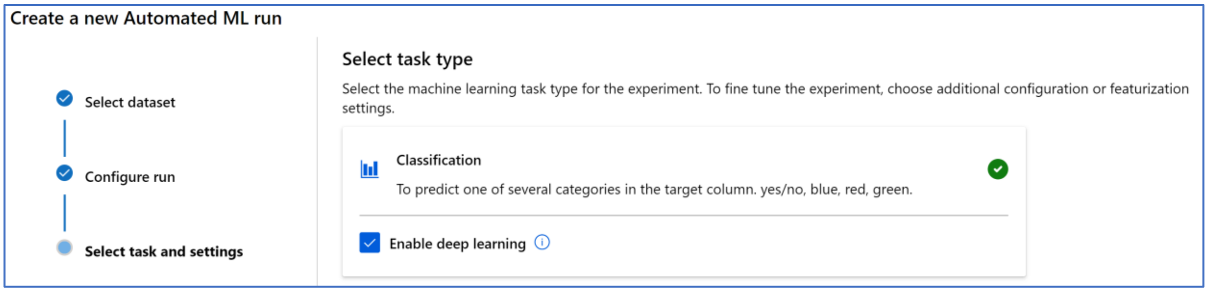
   This takes a couple minutes to complete.

4. After creation, select your new compute target from the drop-down list.

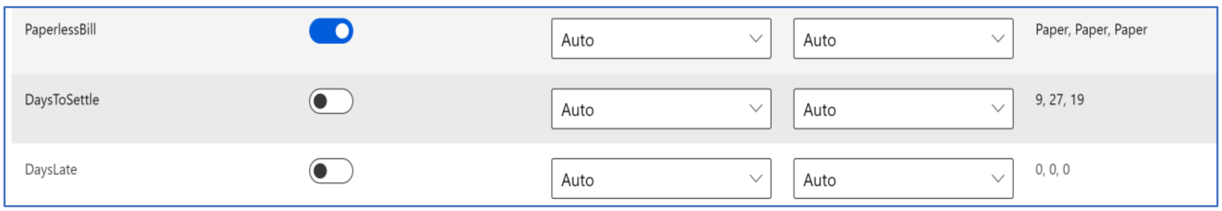
TABLE 5

Additional configurations	Description	Value for tutorial
Primary metric	Evaluation metric that the machine learning algorithm will be measured by.	AUC_weighted
Explain best model	Automatically shows explainability on the best model created by automated ML.	Enable
Blocked algorithms	Algorithms you want to exclude from the training job	None
Exit criterion	If a criteria is met, the training job is stopped.	Training job time (hours): 1 Metric score threshold: None
Validation	Choose a cross-validation type and number of tests.	Validation type: k-fold cross-validation Number of validations: 2
Concurrency	The maximum number of parallel iterations executed per iteration	Max concurrent iterations: 5

   2. Select **Next**.
3. On the **Task type and settings** form, complete the setup for your automated ML experiment by specifying the machine learning task type and configuration settings.
   1. Select **Classification** as the machine learning task type. Click on Enable Deep Learning



2. Select **View additional configuration settings** and populate the fields as follows. These settings are to better control the training job. Otherwise, defaults are applied based on experiment selection and data.
3. Select **Save**.
4. Select **View featurization settings**. For this example, select the toggle switch for the **"DaysToSettle" and "DaysLate"** feature so as to not include it for featurization in this experiment.



Select **Save**.

4. Select **Finish** to run the experiment. The **Run Detail** screen opens with the **Run status** at the top as the experiment preparation begins. This status updates as the experiment progresses. Notifications also appear in the top right corner of the studio, to inform you of the status of your experiment.

Important

Preparation takes **10-15 minutes** to prepare the experiment run. Once running, it takes **2-3 minutes more for each iteration**.

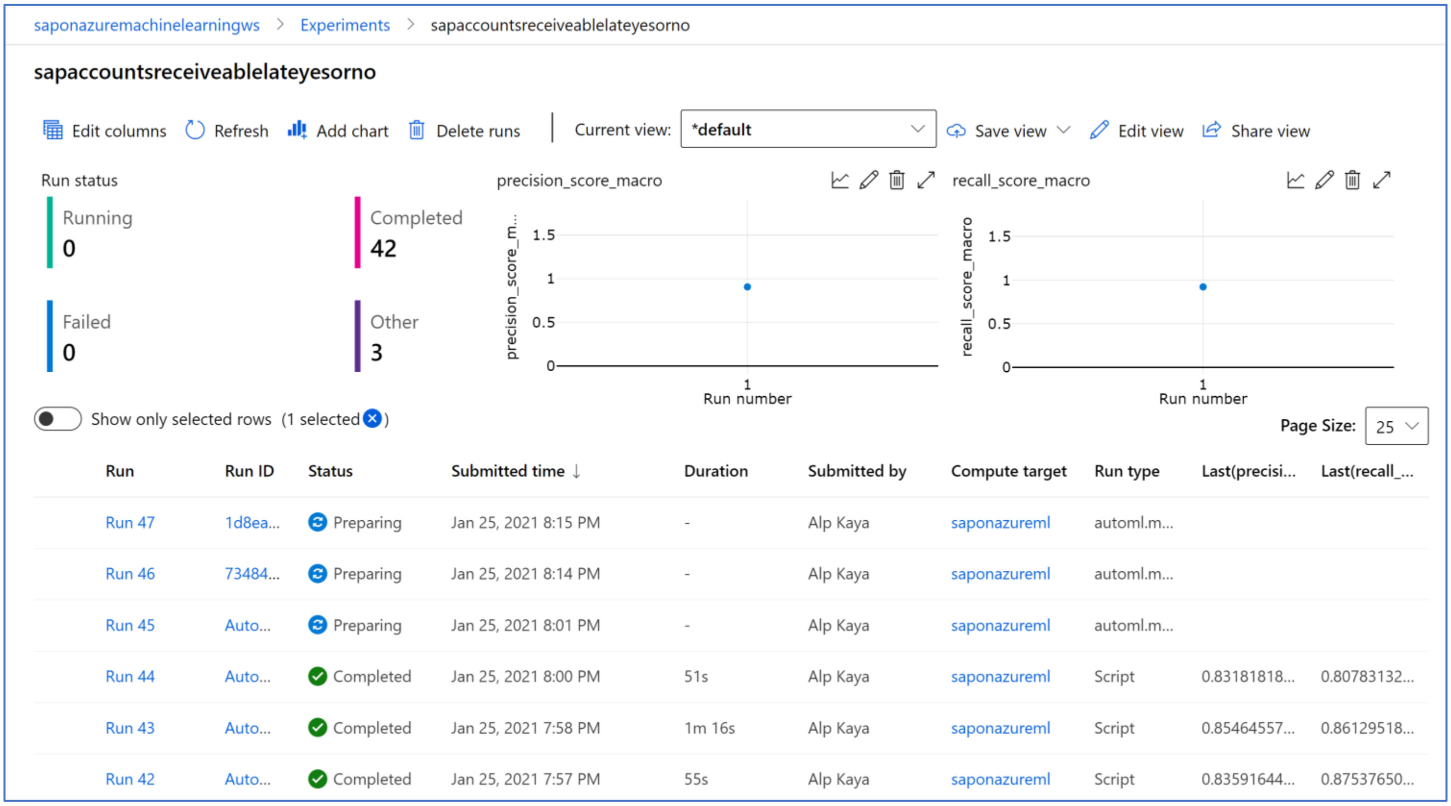
In production, you'd likely walk away for a bit. But for this tutorial, we suggest you start exploring the tested algorithms on the **Models** tab as they complete while the others are still running.

Explore models

Navigate to the **Models** tab to see the algorithms (models) tested. By default, the models are ordered by metric score as they complete. For this tutorial, the model that scores the highest based on the chosen **AUC_weighted** metric is at the top of the list.

While you wait for all of the experiment models to finish, select the **Algorithm name** of a completed model to explore its performance details.

The following navigates through the **Details** and the **Metrics** tabs to view the selected model's properties, metrics, and performance charts.



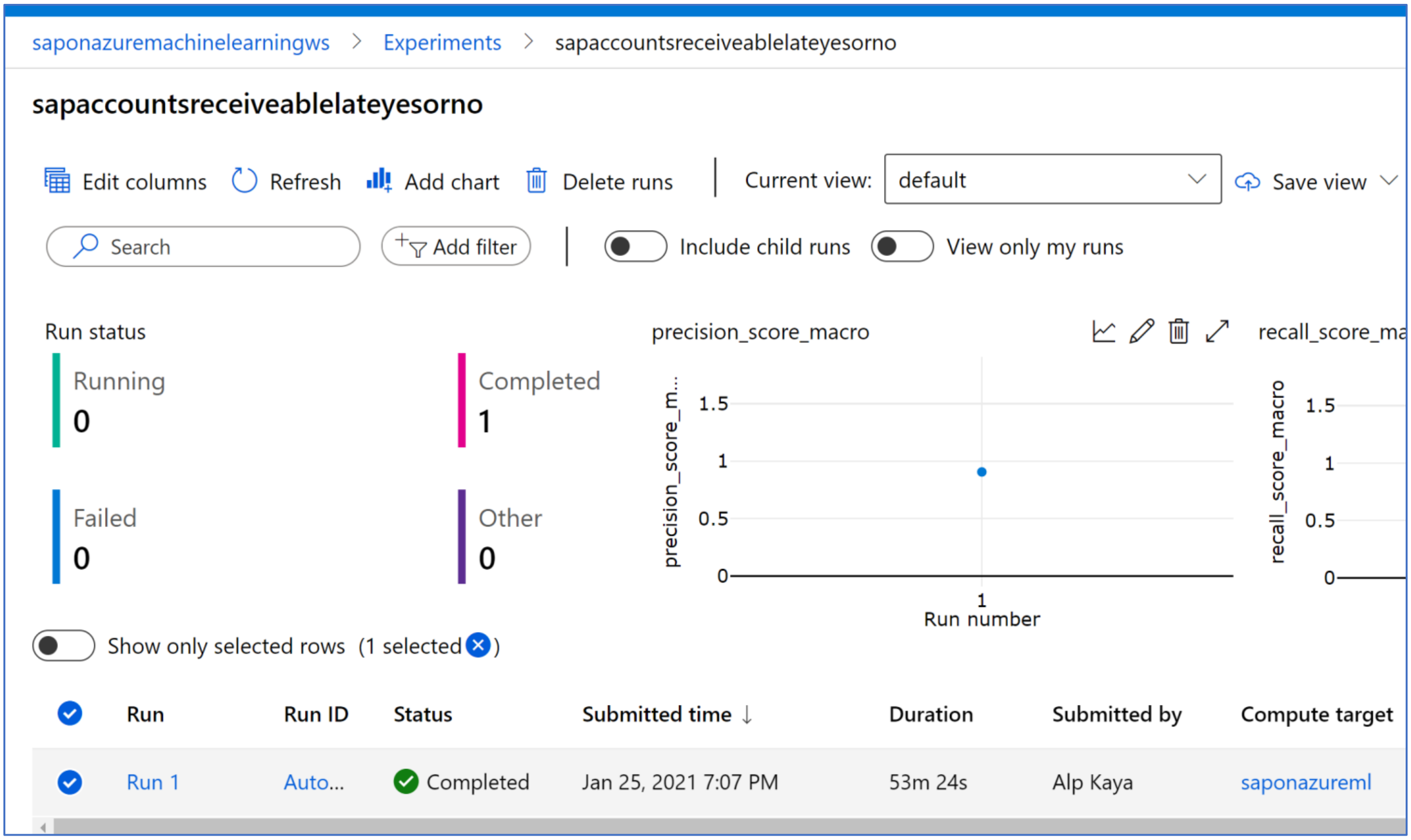
You can see dozens and dozens of models were executed. Several algorithms were selected and several parameters tuned with subsequent executions in an intelligent manner. If you select any particular Run/Run ID, you can now deploy your model into a container where it will be exposed as a REST API for consumption on new data.

Deploy Best Models

The automated machine learning interface allows you to deploy the best model as a web service in a few steps. Deployment is the integration of the model so it can predict on new data and identify potential areas of opportunity.

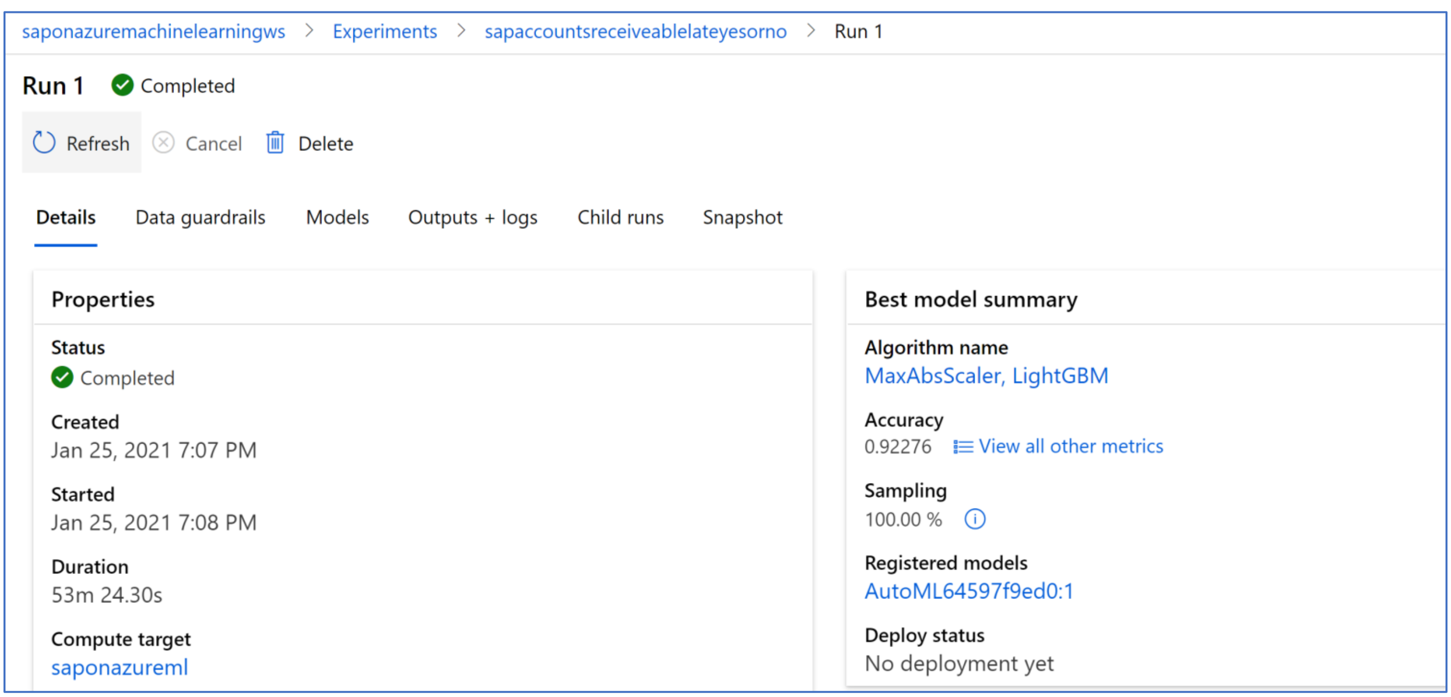
For this experiment, deployment to a web service means that you now have an iterative and scalable web solution for identifying if new accounts receivables will be paid on time or not.

Check to see if your experiment run is complete. To do so, navigate back to the parent run page by selecting **Run 1** at the top of your screen. A **Completed** status is shown on the top left of the screen.



Click on Run 1 above

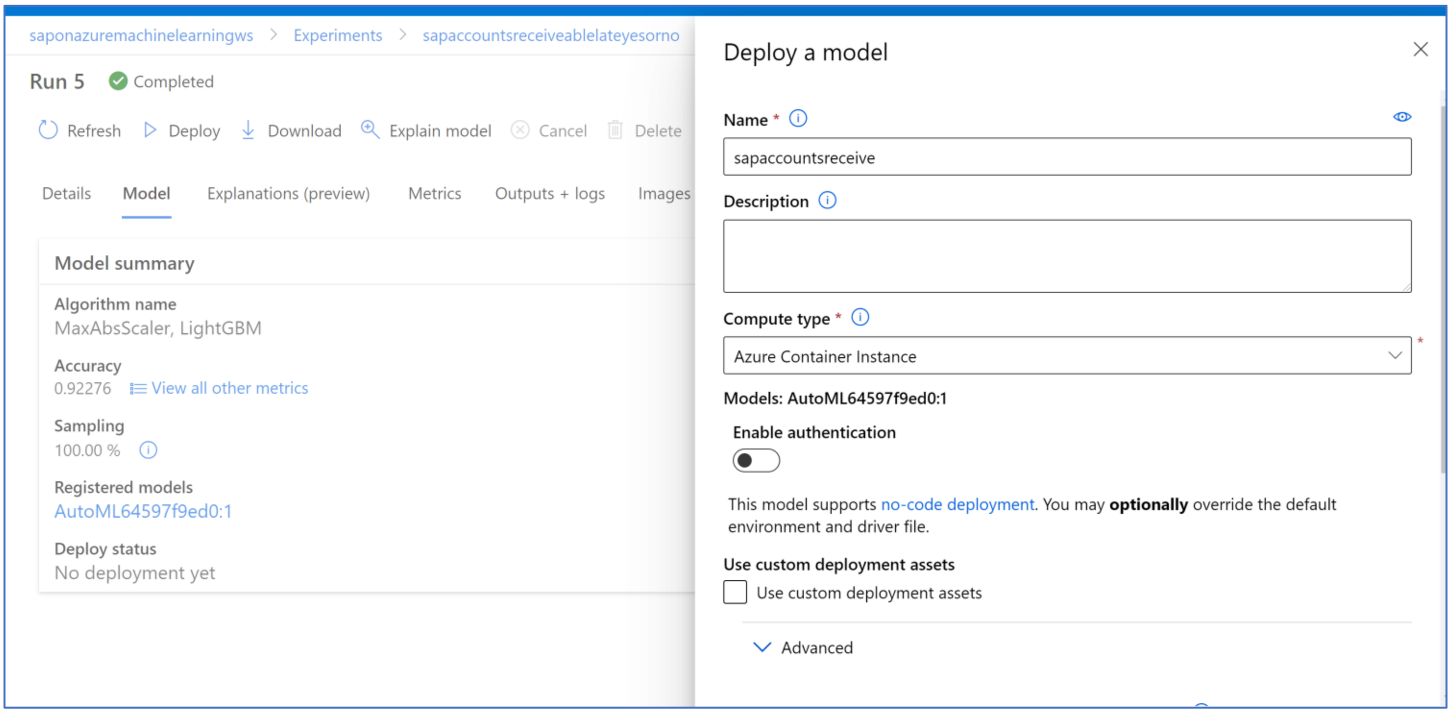
You can see the best model is the **MaxAbsScalar** and **LightGBM.** Click on **MaxAbsScalar and LIghtGBM**



Click on **Deploy a model** pane as follows:

TABLE 6

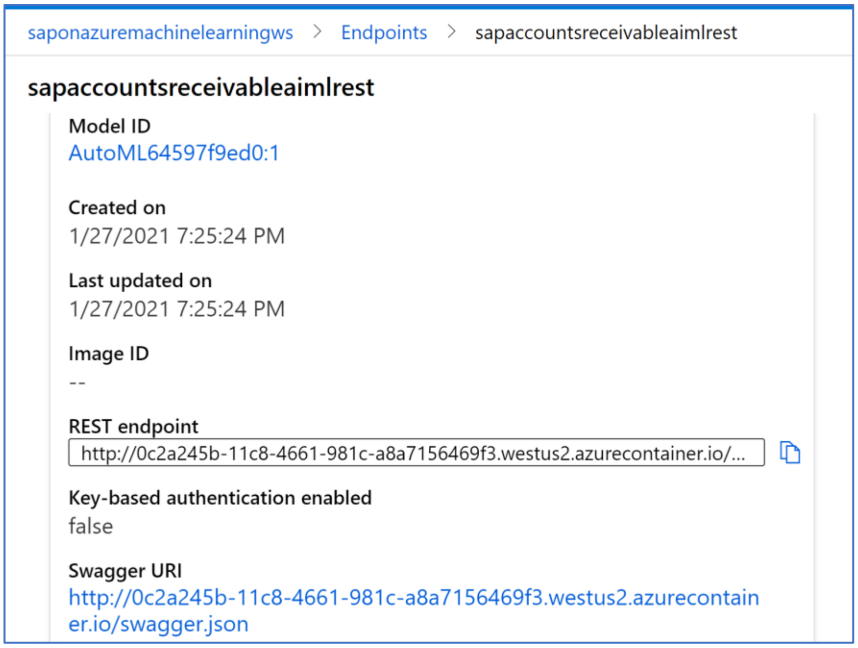
Field	Value
Deployment name	sapaccountsreceive
Deployment description	My first automated machine learning experiment deployment
Compute type	Select Azure Compute Instance (ACI)
Enable authentication	Disable.
Use custom deployments	Disable. Allows for the default driver file (scoring script) and environment file to be autogenerated.



1. For this example, we use the defaults provided in the *Advanced* menu.
2. Select **Deploy**.

   A green success message appears at the top of the **Run** screen, and in the **Model summary** pane, a status message appears under **Deploy status**.
   Select **Refresh** periodically to check the deployment status.

Now you have an operational web service to generate predictions.



REST endpoint:

http://0c2a245b-11c8-4661-981c-a8a7156469f3.westus2.azurecontainer.io/score

Swagger URI:

http://0c2a245b-11c8-4661-981c-a8a7156469f3.westus2.azurecontainer.io/swagger.json

You can now submit new Accounts Receivables transaction to this end point and get a response predicant if the customer will pay on time (or not)