# Innovate and improve your business with Azure Conversational AI and Synapse Analytics

## Contents

Create An Azure Synapse Workspace ........ 1

Configure the Synapse Workspace ........ 3

Create New Dedicated SQL Pool ........ 3

Create New Apache Spark pool ........ 3

Link Azure Machine Learning and Synapse Workspaces ........ 4

Link Azure Machine Learning Workspace to Synapse Workspace ........ 4

Link Synapse Workspace to Azure Machine Learning Workspace ........ 5

Consume Data in Azure Synapse ........ 8

Explore Data in Azure Synapse ........ 12

Execute a Basic SQL Script ........ 12

Explore Relationships Between Variables ........ 13

Launch a Regression Model in Azure Synapse ........ 17

Train the Regression Model ........ 17

Launch a Classification Model in Azure Synapse ........ 20

Train the Classification Model ........ 20

Launch a Time Series Model in Azure Synapse ........ 23

Train the Time Series Model ........ 23

## Data Insight Highlights

1. The general trends in different conversation types over time (telephony increasing while others go down)
2. Conversations with higher ratings tended to have longer durations
3. Conversations with positive sentiment analysis tended to have longer conversation durations, higher conversation ratings, and higher brand ratings

## Create An Azure Synapse Workspace

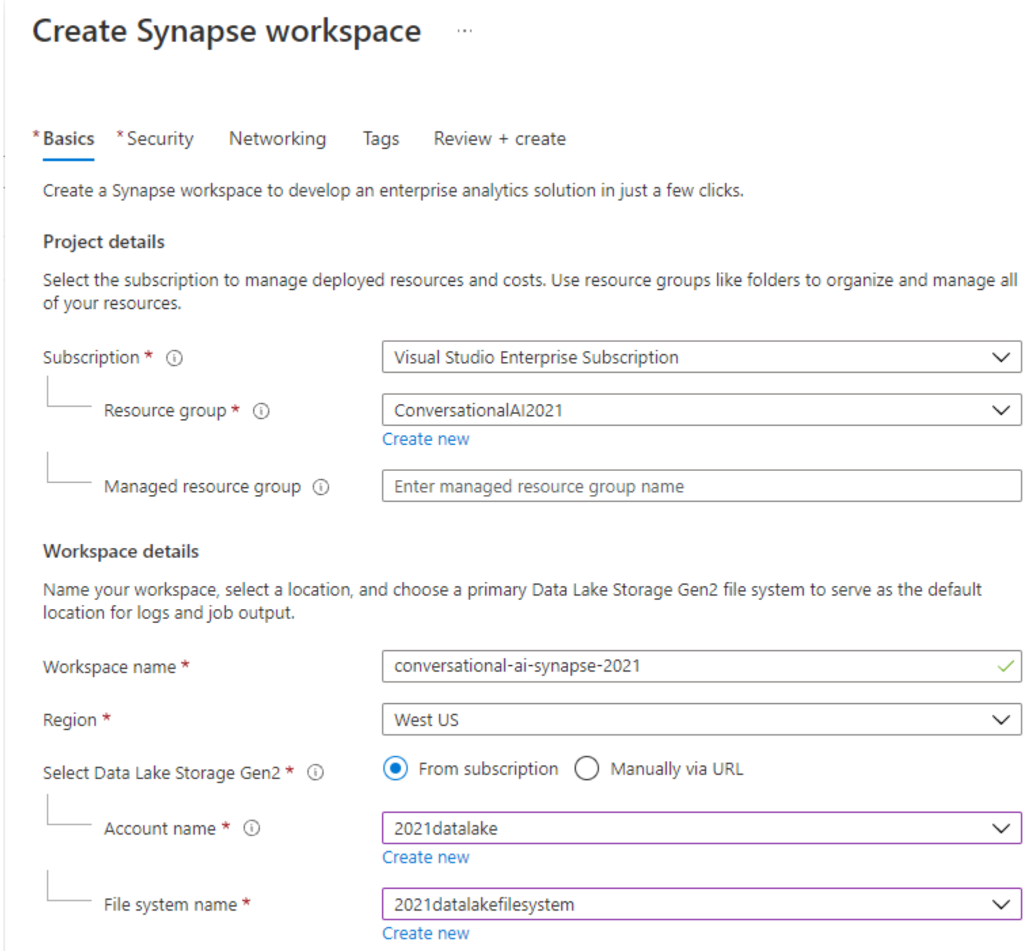
Before we can start working with Azure Synapse, we need to do set up our Synapse Workspace.

1. Sign into the Azure Portal
2. Search for “Azure Synapse Analytics” to navigate to the setup page.
3. Click “Create” to start creating your workspace.

From this page, you can configure your workspace:

1. Select your subscription from the subscription menu.
2. For the resource group, you can select the resource group that you created for the Azure Machine Learning Workspace.
3. For the workspace name, fill in the name of your new workspace.
4. If needed, update the region to match your local timezone.
5. For the account name, click “Create New” and enter an account name for the workspace. We called ours “2021datalake.”
6. For the file system name, click “Create New” and enter a file system name for the workspace. We called ours “2021datalakefilesystem.”

Your settings should look similar to this:



Create Synapse workspace

Create a Synapse workspace to develop an enterprise analytics solution in just a few clicks.

Project details

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all of your resources.

| | |
|---|---|
| Subscription * | Visual Studio Enterprise Subscription |
| Resource group * | ConversationalAI2021<br>Create new |
| Managed resource group | Enter managed resource group name |

Workspace details

Name your workspace, select a location, and choose a primary Data Lake Storage Gen2 file system to serve as the default location for logs and job output.

| | |
|---|---|
| Workspace name * | conversational-ai-synapse-2021 ✓ |
| Region * | West US |
| Select Data Lake Storage Gen2 * | ● From subscription ○ Manually via URL |
| Account name * | 2021datalake<br>Create new |
| File system name * | 2021datalakefilesystem<br>Create new |

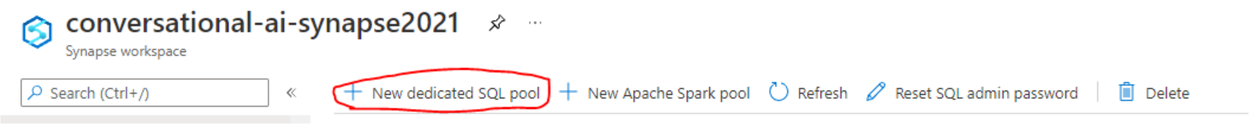
Now, you can click “Review + create” and your workspace’s settings will be validated. Once validation has passed, click “Create” on the following page to deploy the workspace. This may take a few minutes to complete.

# Configure the Synapse Workspace

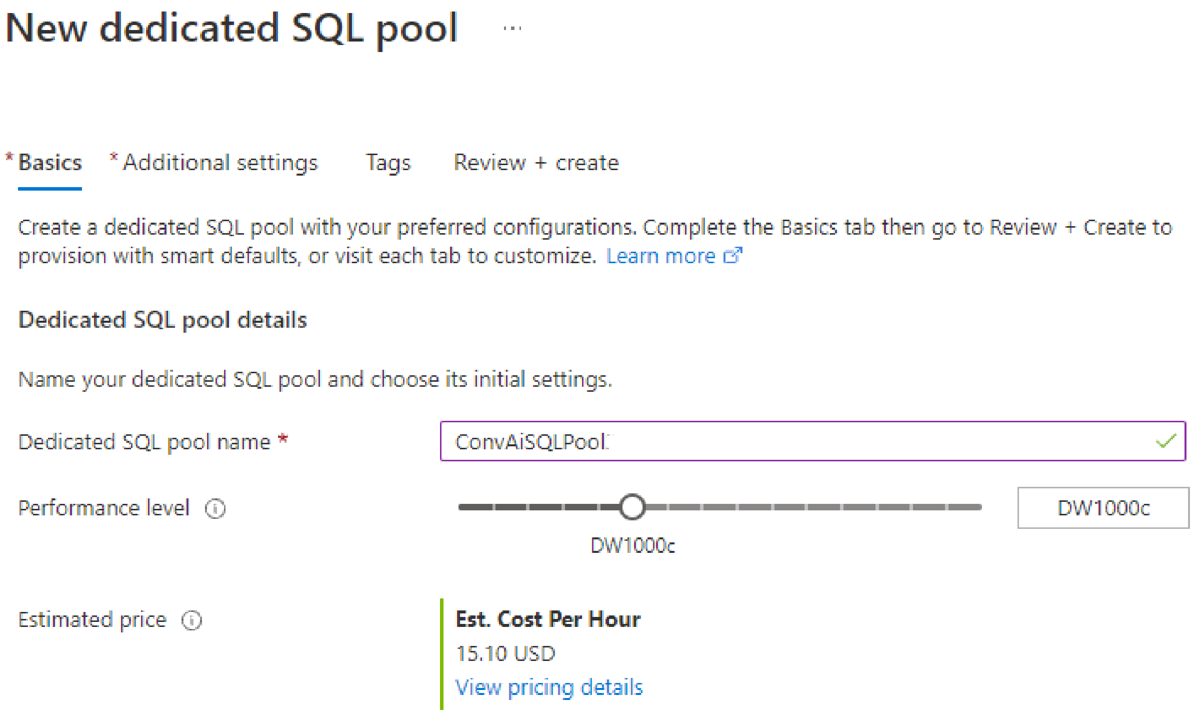
Once deployment is complete, you can access your workspace through the Azure Portal. In the search bar, search for the name of your Synapse workspace. In our case, we'll navigate to "conversational-ai-synapse2021."

## Create New Dedicated SQL Pool

From the main Synapse workspace page, click "New dedicated SQL pool"



Enter a name for the Dedicated SQL Pool. The other values can be left on their default settings:



New dedicated SQL pool ...

* Basics * Additional settings Tags Review + create

Create a dedicated SQL pool with your preferred configurations. Complete the Basics tab then go to Review + Create to provision with smart defaults, or visit each tab to customize. Learn more

Dedicated SQL pool details

Name your dedicated SQL pool and choose its initial settings.

Dedicated SQL pool name * ConvAiSQLPool ✓

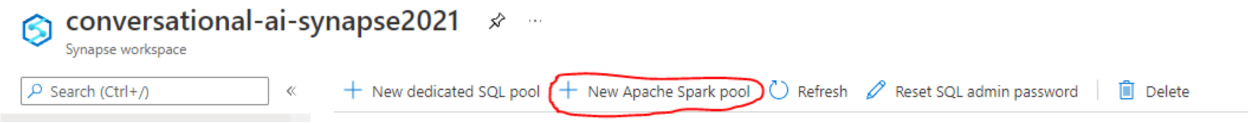
Performance level ⓘ DW1000c DW1000c

Estimated price ⓘ **Est. Cost Per Hour** 15.10 USD View pricing details

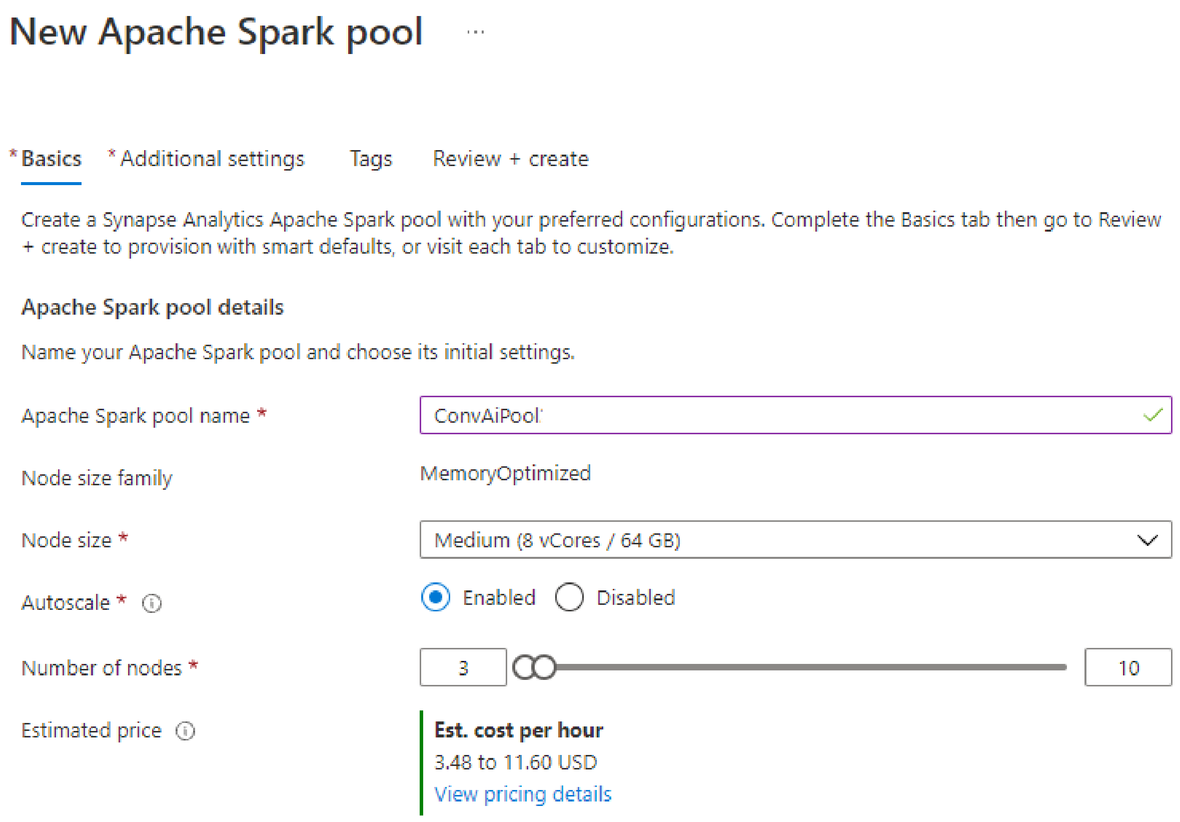
Select "Review + Create" to validate the SQL pool settings. Once validation has passed, click "Create" to begin deployment of the new SQL pool.

## Create New Apache Spark pool

From the main Synapse workspace page, click "New Apache Spark pool"



Enter a name for the Apache Spark pool. The other values can be left on their default settings:



New Apache Spark pool ···

* Basics * Additional settings Tags Review + create

Create a Synapse Analytics Apache Spark pool with your preferred configurations. Complete the Basics tab then go to Review + create to provision with smart defaults, or visit each tab to customize.

Apache Spark pool details

Name your Apache Spark pool and choose its initial settings.

Apache Spark pool name * ConvAiPool ✓

Node size family MemoryOptimized

Node size * Medium (8 vCores / 64 GB)

Autoscale * ● Enabled ○ Disabled

Number of nodes * 3 10

Estimated price **Est. cost per hour** 3.48 to 11.60 USD View pricing details

Select “Review + Create” to validate the Apache Spark pool settings. Once validation has passed, click “Create” to begin deployment of the new Apache Spark pool.

# Link Azure Machine Learning and Synapse Workspaces

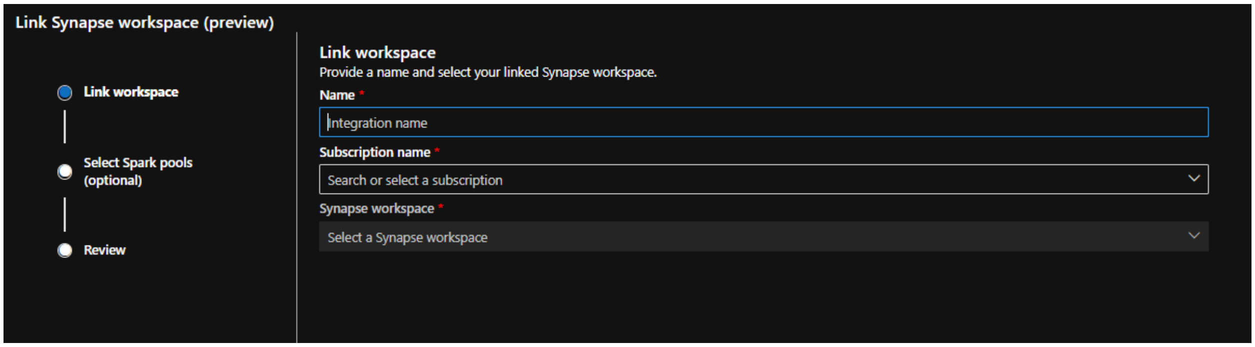
Now we’re ready to link the Azure Machine Learning workspace to the Synapse workspace and vice versa.

## Link Azure Machine Learning Workspace to Synapse Workspace

You’re now ready to link your Machine Learning workspace to your Synapse workspace. This step allows you to access the Synapse workspace from within the Azure Machine Learning workspace.

1. Navigate to the Azure Machine Learning Studio to access your existing Machine Learning workspace.
2. In the left-hand panel, select “Linked Services.”
3. Click “Add Integration”

From here, you should name your Linked Workspace, select the subscription that your Synapse workspace is connected to, and select your Synapse workspace. In our case, we named our Synapse workspace ml-synapse-link:


Link Synapse workspace (preview)
Link workspace
Select Spark pools (optional)
Review

Link workspace

Provide a name and select your linked Synapse workspace.

Name *: Integration name

Subscription name *: Search or select a subscription

Synapse workspace *: Select a Synapse workspace

After updating the settings click Next.

On the next screen, select the Apache Spark pool that you created in the previous step.

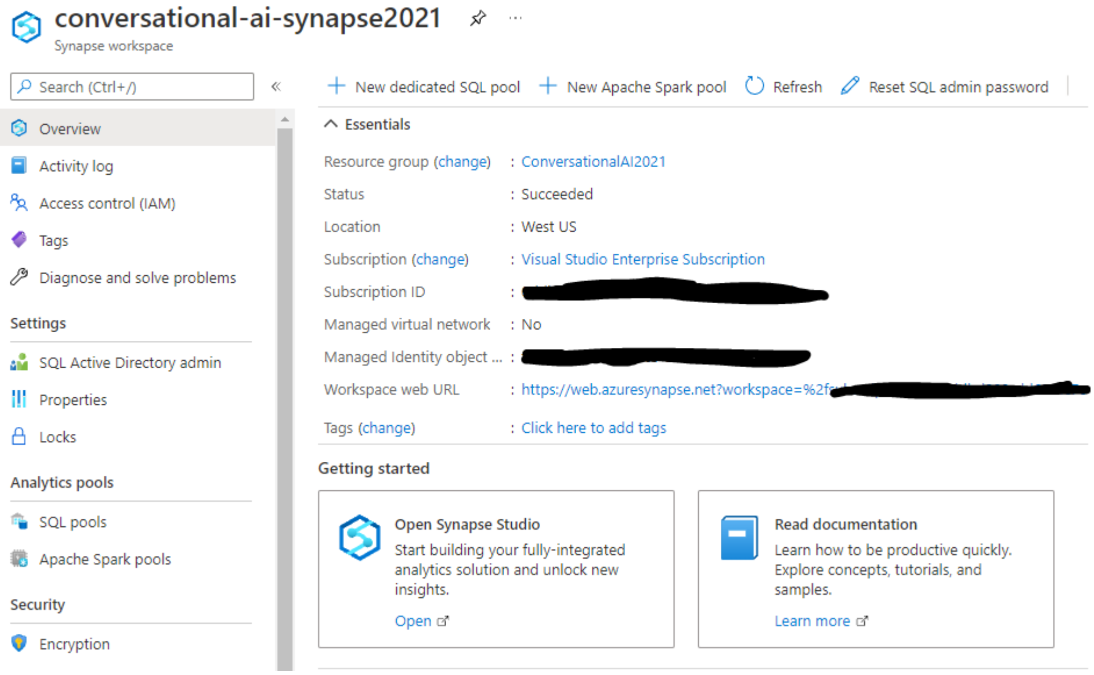
After reviewing your settings, click “Create” to finish linking the workspaces.

## Link Synapse Workspace to Azure Machine Learning Workspace

Now, you can link your Synapse workspace to your Machine Learning workspace. This step allows you to access the Azure Machine Learning workspace from the Synapse workspace.

Navigate back to the [Azure Portal](). In the search bar, search for the name of your Synapse workspace. In this example, we’ll navigate to “conversational-ai-synapse2021.”

To open your Azure Synapse workspace, click the Workspace web URL or click “Open Synapse Studio”:



| | |
|---|---|
| Resource group (change) | : ConversationalAI2021 |
| Status | : Succeeded |
| Location | : West US |
| Subscription (change) | : Visual Studio Enterprise Subscription |
| Subscription ID | : [redacted] |
| Managed virtual network | : No |
| Managed Identity object ... | : [redacted] |
| Workspace web URL | : https://web.azuresynapse.net?workspace=%2f [redacted] |
| Tags (change) | : Click here to add tags |

Getting started

Open Synapse Studio

Start building your fully-integrated analytics solution and unlock new insights.

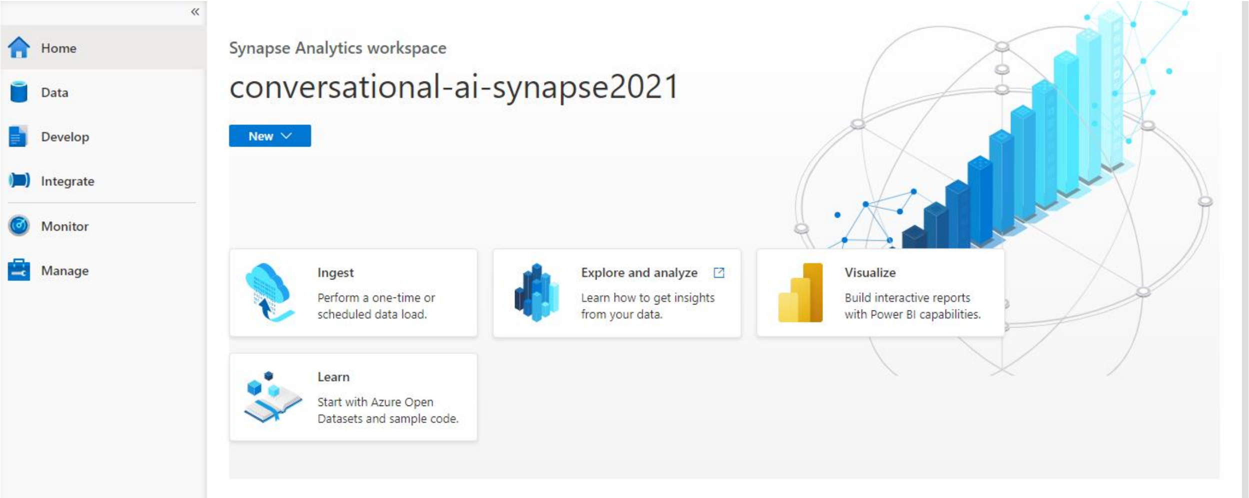
Open

Read documentation

Learn how to be productive quickly. Explore concepts, tutorials, and samples.

Learn more

The Azure Synapse page will look like this:



Synapse Analytics workspace

conversational-ai-synapse2021

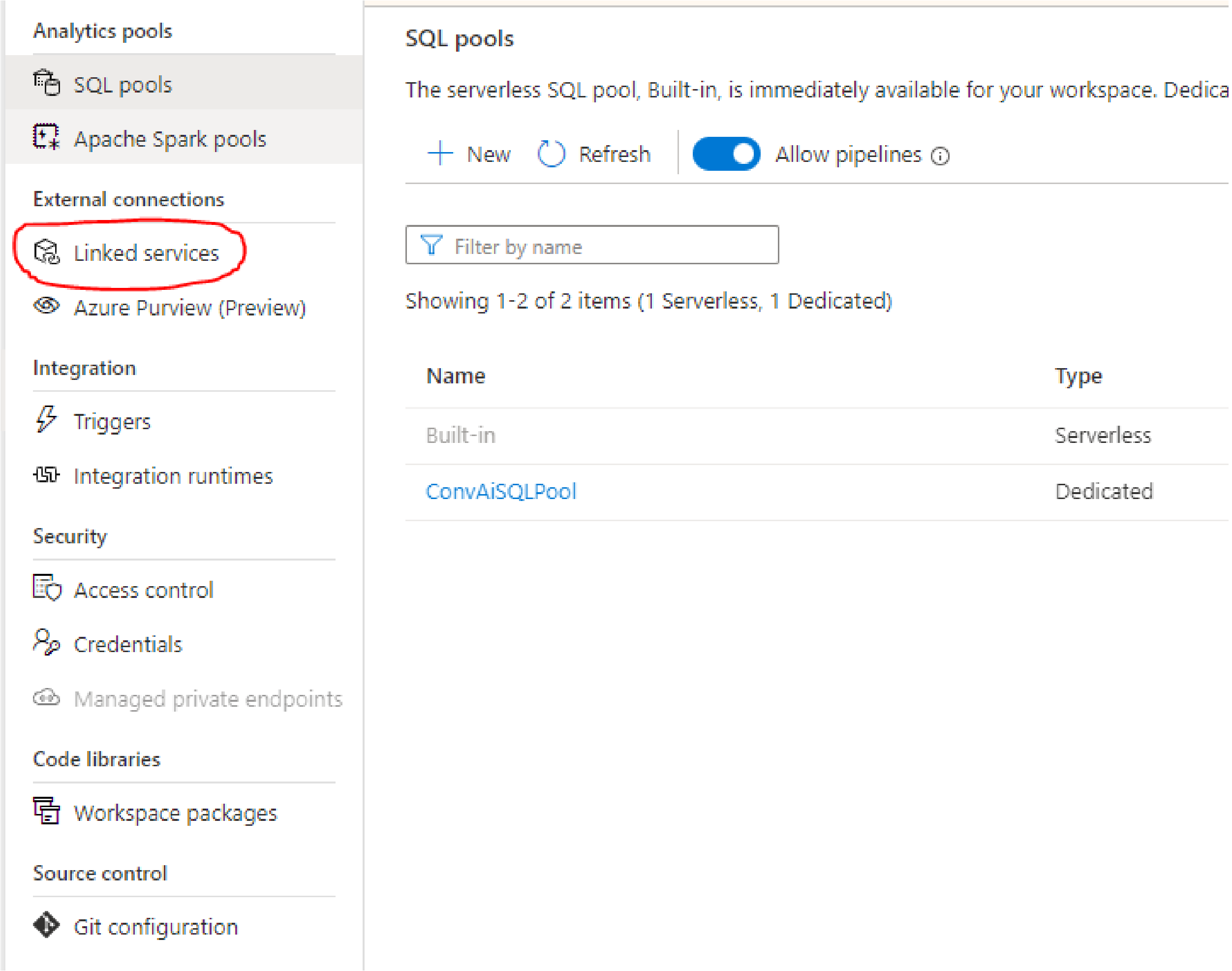
Ingest
Perform a one-time or scheduled data load.

Explore and analyze
Learn how to get insights from your data.

Visualize
Build interactive reports with Power BI capabilities.

Learn
Start with Azure Open Datasets and sample code.

In the left-hand panel, select the “Manage” button. Then, click “Linked Services” in the new panel:



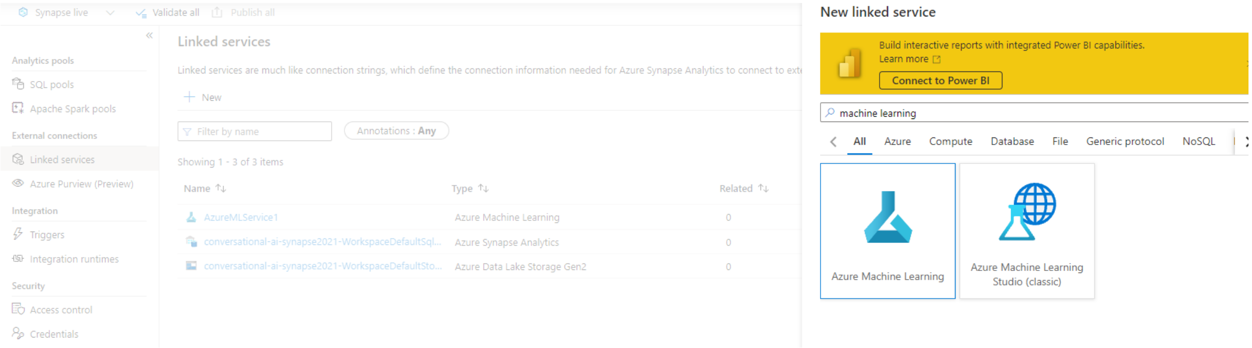
SQL pools

The serverless SQL pool, Built-in, is immediately available for your workspace. Dedica

Showing 1-2 of 2 items (1 Serverless, 1 Dedicated)

| Name | Type |
|---|---|
| Built-in | Serverless |
| ConvAiSQLPool | Dedicated |

On the Linked Services page, click “New.” Then, search for “Azure Machine Learning.” Click Continue:



Linked services

Linked services are much like connection strings, which define the connection information needed for Azure Synapse Analytics to connect to exte

Showing 1 - 3 of 3 items

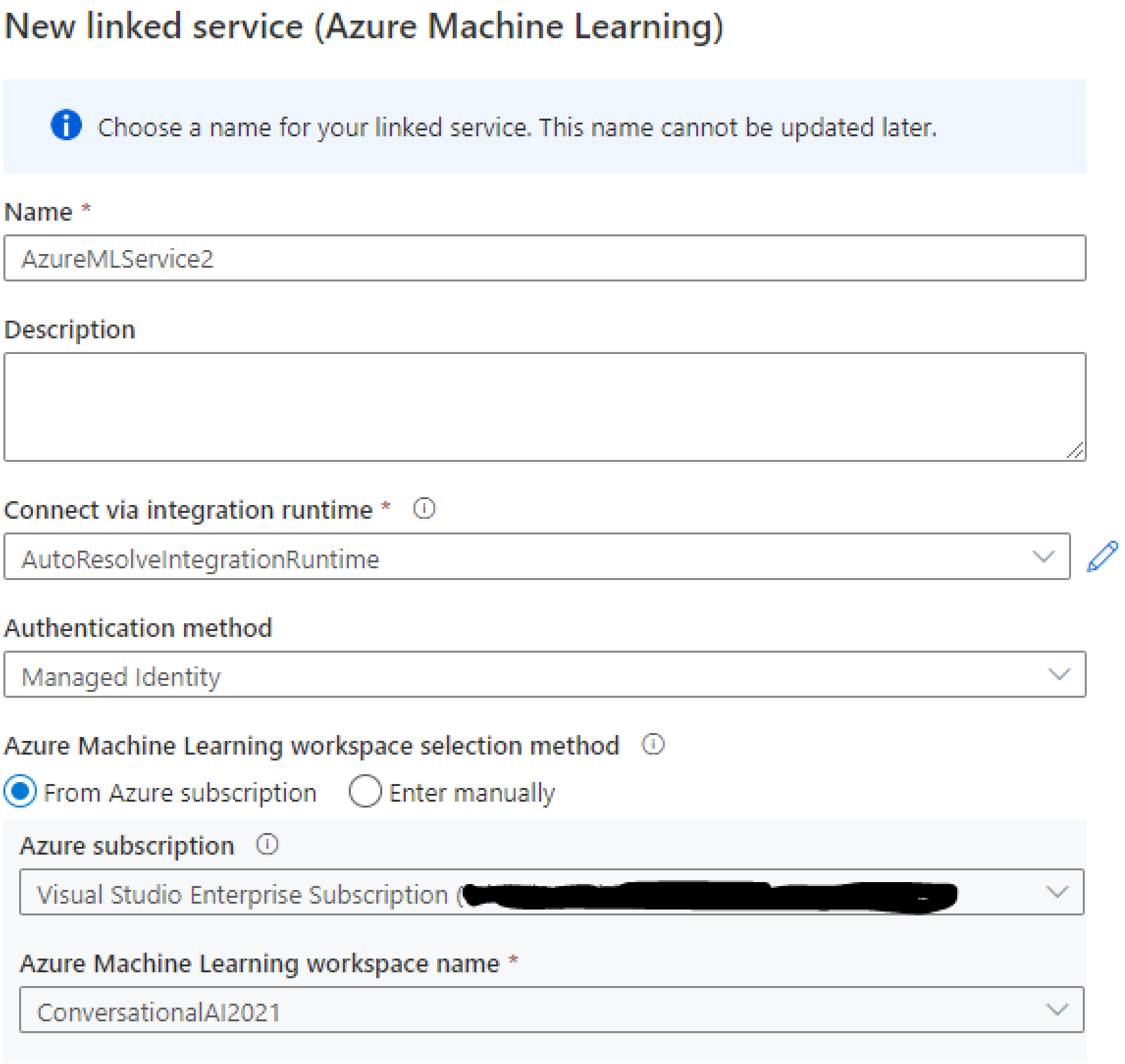
| Name ↑↓ | Type ↑↓ | Related ↑↓ |
|---|---|---|
| AzureMLService1 | Azure Machine Learning | 0 |
| conversational-ai-synapse2021-WorkspaceDefaultSql... | Azure Synapse Analytics | 0 |
| conversational-ai-synapse2021-WorkspaceDefaultSto... | Azure Data Lake Storage Gen2 | 0 |

New linked service

Build interactive reports with integrated Power BI capabilities.
Learn more

machine learning

Name your Linked Service, select the subscription that your Azure Machine Learning workspace is connected to, and select your Machine Learning workspace. Your settings should look something like this:



New linked service (Azure Machine Learning)

Choose a name for your linked service. This name cannot be updated later.

Name *
AzureMLService2

Description

Connect via integration runtime *
AutoResolveIntegrationRuntime

Authentication method
Managed Identity

Azure Machine Learning workspace selection method
● From Azure subscription ○ Enter manually

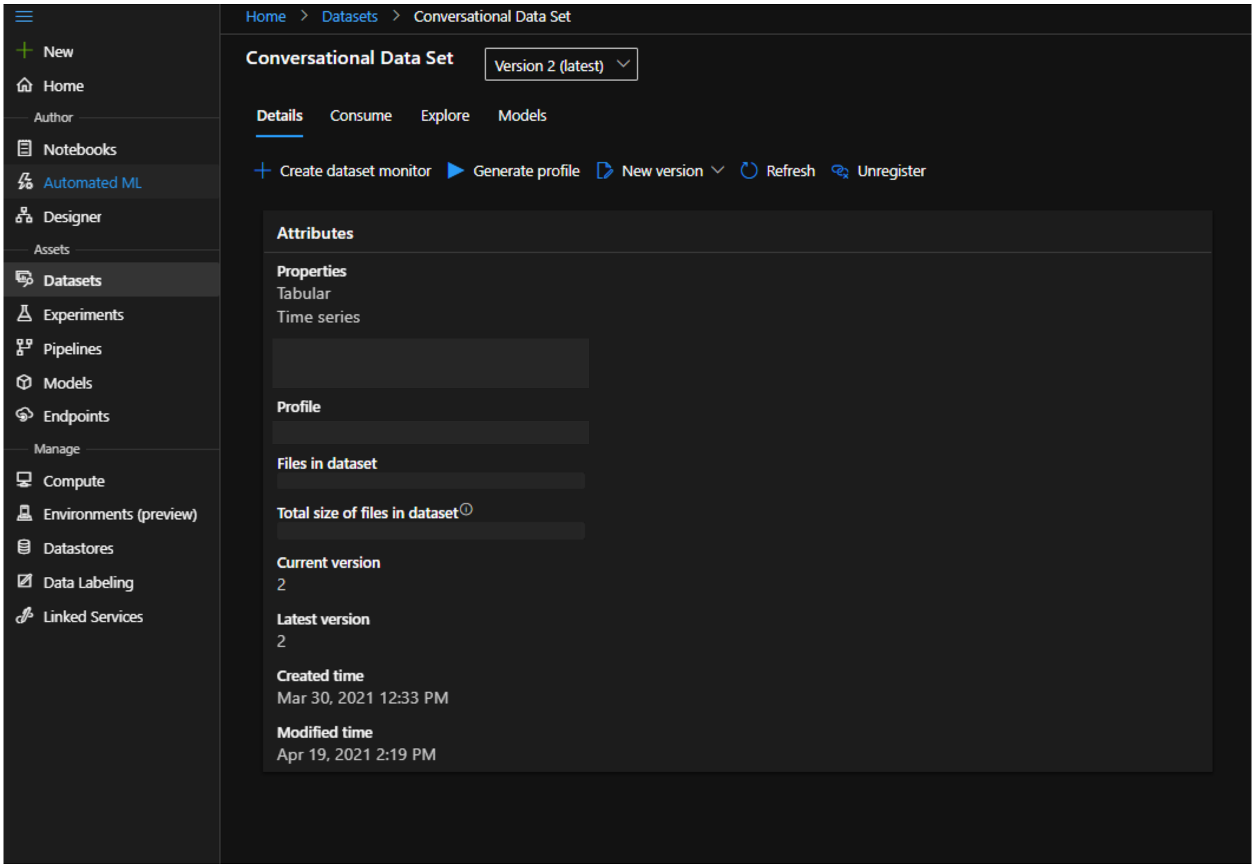
Azure subscription
Visual Studio Enterprise Subscription ([redacted]

Azure Machine Learning workspace name *
ConversationalAI2021

Click “Create” to link the workspaces. Then, be sure to click “Publish All” to finalize the linked workspaces.

# Consume Data in Azure Synapse

In another window, navigate back to the Azure Machine Learning Studio to access your existing Machine Learning workspace. Navigate to the dataset you'd like to import. In our case, we'll start with the initial Conversational Data Set:



Conversational Data Set

Attributes

Properties
Tabular
Time series

Profile

Files in dataset

Total size of files in dataset

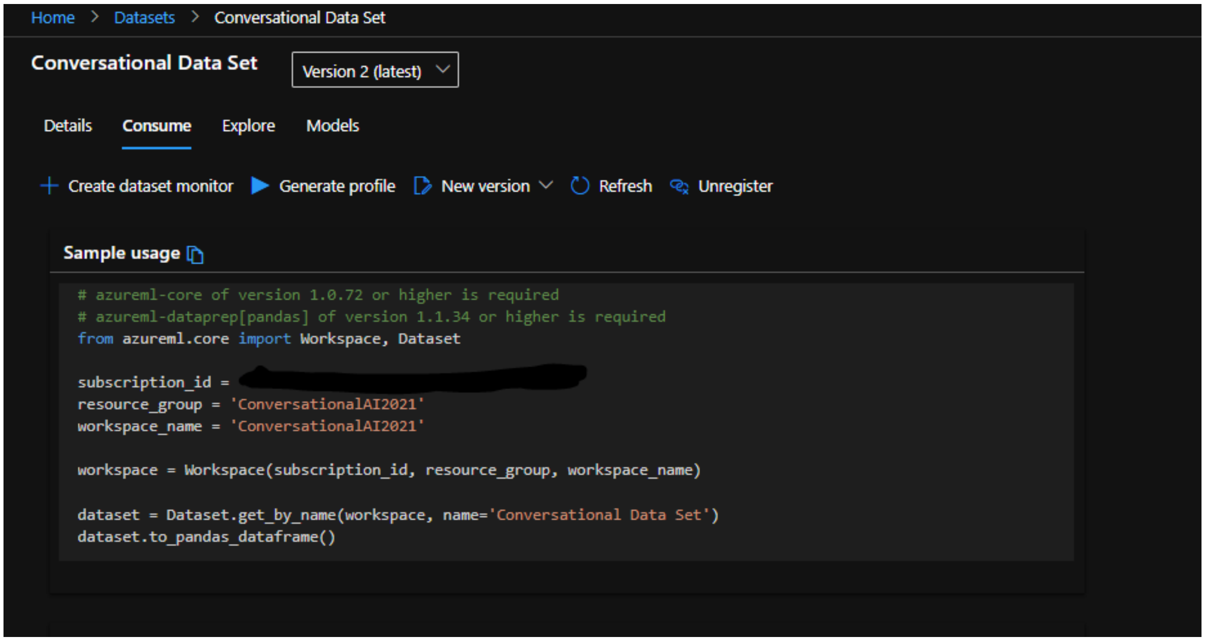
Current version
2

Latest version
2

Created time
Mar 30, 2021 12:33 PM

Modified time
Apr 19, 2021 2:19 PM

Navigate to the "Consume" tab to find the code necessary to import your data into the Azure Synapse workspace:

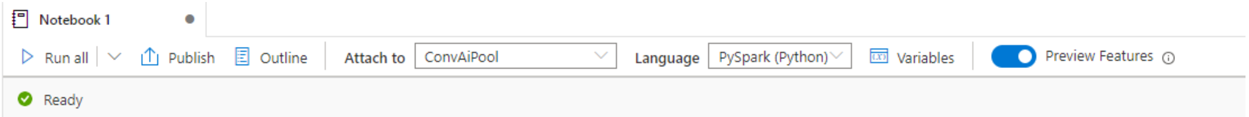


```
# azureml-core of version 1.0.72 or higher is required
# azureml-dataprep[pandas] of version 1.1.34 or higher is required
from azureml.core import Workspace, Dataset

subscription_id = [redacted]
resource_group = 'ConversationalAI2021'
workspace_name = 'ConversationalAI2021'

workspace = Workspace(subscription_id, resource_group, workspace_name)

dataset = Dataset.get_by_name(workspace, name='Conversational Data Set')
dataset.to_pandas_dataframe()
```

Now, we can create a Jupyter notebook in our Azure Synapse workspace to import this data.

1. Navigate back to the Azure Synapse workspace.
2. Select “Develop in the left-hand panel.”
3. Click the “+” icon, and then click “Notebook” to generate a new Jupyter notebook.

Once you’ve launched the notebook, set the “Attach To” field to the Apache Spark Pool created earlier:



Attach to ConvAiPool
Language PySpark (Python)
Ready

Now, you can paste text from your ML Azure workspace into the first code block. This should look something like this:

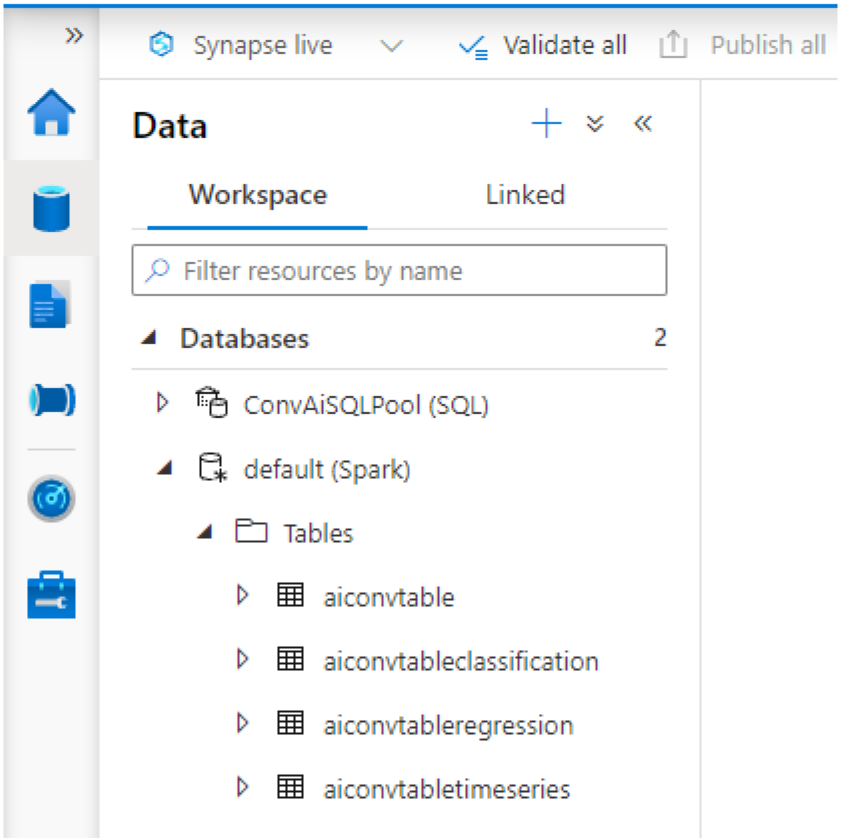
```python
from azureml.core import Workspace, Dataset

subscription_id = YOUR_SUBCRIPTION_ID_HERE
resource_group = 'ConversationalAI2021'
workspace_name = 'ConversationalAI2021'

workspace = Workspace(subscription_id, resource_group, workspace_name)

dataset = Dataset.get_by_name(workspace, name='Conversational Data Set')
dataset.to_pandas_dataframe()
```

This data needs to be written to a table in order to initiate a ML model. From Azure Synapse. To do this, we can add the following code in the next code block:

``` python
df = dataset.to_pandas_dataframe()
df.columns = df.columns.str.replace(' ','_') # Update names so they're valid in SQL

spark_df = spark.createDataFrame(df)
spark_df.write.mode("overwrite").saveAsTable("default.AiConvTable")
```

Click “Run All” to execute the code in your Jupyter notebook. This should take a few minutes to run.

To view the data that you’ve imported, navigate to the “Data” icon in the left-hand panel. You can then find the table under the default database:



We can then repeat this process for the datasets we created for training the models in the Machine Learning workspace. For simplicity, I’ve condensed this into one Jupyter notebook. you should also be able to access it in Synapse directly through Develop -> Notebooks -> ConvAiNotebook

```python

from azureml.core import Workspace, Dataset

# Set this to your own subscription ID
subscription_id = YOUR_SUBSCRIPTION_ID
# Set this to your Azure ML resource group
resource_group = 'ConversationalAI2021'
# Set this to your Azure ML workspace name
```

```
workspace_name = 'ConversationalAI2021'


# Set these to the names of the input datasets in Azure Machine Learning studio
raw_data = 'Conversational AI Data'
classification_data = 'Classification AI with Success Variable'
regression_data = 'Conversational AI Data'
time_series_data = 'Time-series-data'

# Names of tables to be created in Azure Synapse
raw_table = 'default.AIConvTable'
classification_table = 'default.AiConvTableClassification'
regression_table = 'default.AIConvTableRegression'
time_series_table = 'default.AiConvTableTimeSeries'

workspace = Workspace(subscription_id, resource_group, workspace_name)

def get_data(workspace, dataset_name):
    dataset = Dataset.get_by_name(workspace, name=dataset_name)
    return dataset.to_pandas_dataframe()

def clean_dataframe(df):
    df.columns = df.columns.str.replace(' ','_') # Update names so they're valid in SQL
    return df

def write_to_table(df, table_name):
    df = spark.createDataFrame(df)
    df.write.mode("overwrite").saveAsTable(table_name)

# Start by getting raw conversational dataset (all columns)
df = get_data(workspace, raw_data)
df = clean_dataframe(df)
write_to_table(df, raw_table)

# Start by getting regression data
regression_df = get_data(workspace, regression_data)
regression_df = clean_dataframe(regression_df)
del regression_df['Column2'] # Remove this column since we don't need it
write_to_table(regression_df, regression_table)

# Repeat this process for the classification data
classification_df = get_data(workspace, classification_data)
classification_df = clean_dataframe(classification_df)
write_to_table(classification_df,  classification_table)

# Finally, get time series data
time_series_df = get_data(workspace, time_series_data)
time_series_df = clean_dataframe(time_series_df)
write_to_table(time_series_df, time_series_table)
```

After running this script, you navigate to the “Data” icon in the left-hand panel to view all the data you’ve imported. The tables will be listed under the default database.

# Explore Data in Azure Synapse

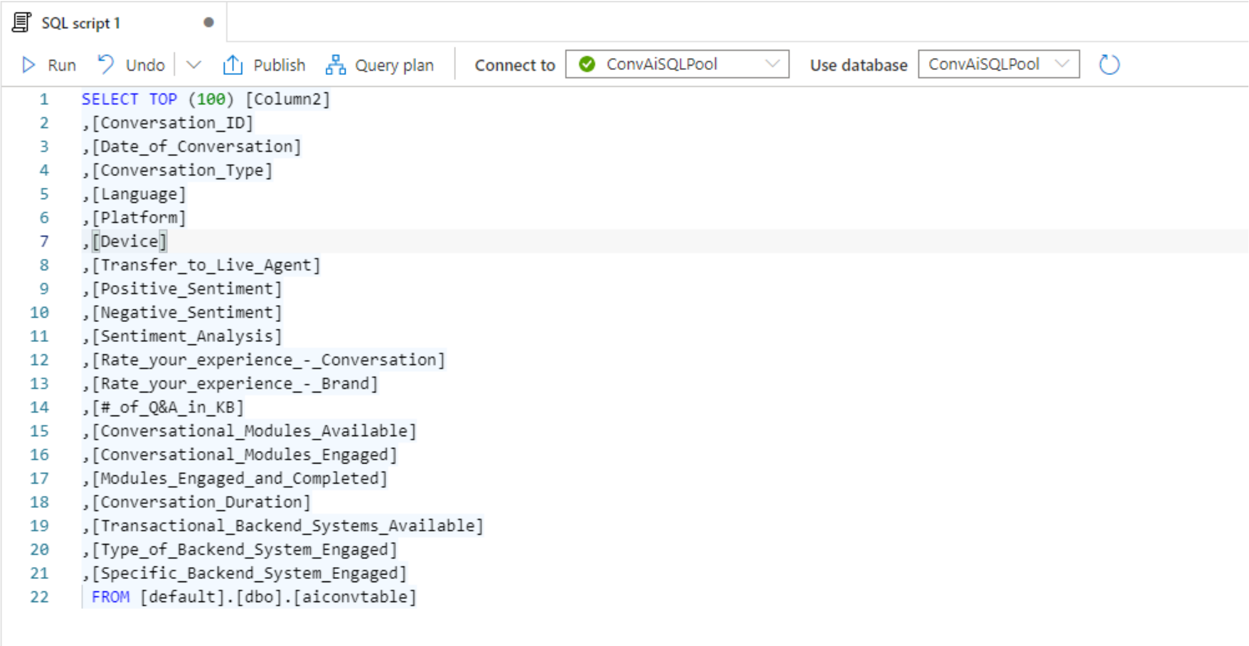
Now that our data has been consumed into Azure Synapse, let’s start by exploring the raw data stored in default.AIConvTable.

## Execute a Basic SQL Script

In the Azure Synapse workspace, navigate to the “Data” option in the left pane. Under Databases, select “default (Spark)” and then “Tables.”

1. Select the data that we will be exploring, in our case “aiconvtable”
2. Select the 3-dot menu next to this table
3. Select “New SQL Script”
4. Select “Select TOP 100 rows”

This will open up a new SQL script that will select the top 100 rows in the data table. This query includes all variables. We’ll name this script “SelectTop100Script.” It should look something like this:



Connect to: ConvAiSQLPool

Use database: ConvAiSQLPool

```sql
SELECT TOP (100) [Column2]
,[Conversation_ID]
,[Date_of_Conversation]
,[Conversation_Type]
,[Language]
,[Platform]
,[Device]
,[Transfer_to_Live_Agent]
,[Positive_Sentiment]
,[Negative_Sentiment]
,[Sentiment_Analysis]
,[Rate_your_experience_-_Conversation]
,[Rate_your_experience_-_Brand]
,[#_of_Q&A_in_KB]
,[Conversational_Modules_Available]
,[Conversational_Modules_Engaged]
,[Modules_Engaged_and_Completed]
,[Conversation_Duration]
,[Transactional_Backend_Systems_Available]
,[Type_of_Backend_System_Engaged]
,[Specific_Backend_System_Engaged]
 FROM [default].[dbo].[aiconvtable]
```

Before we run the script, we need to make a few minor changes:

1. In the bar along the top of the script, set “Connect to” to “ConvAiSQLPool”
2. Update the bottom line of the script to read “FROM [default].[aiconvtable]”

Once you’ve made these changes, select the “Run” button to execute the script. The output will show a table including the top 100 rows from the table. You can also visualize the data by navigating to the “chart” view. Note: The chart view is only available when executing the script in SQL rather than through Jupyter notebooks.

Hit “Publish all” above your script to save your changes so that you can execute the script alone.

## Explore Relationships Between Variables

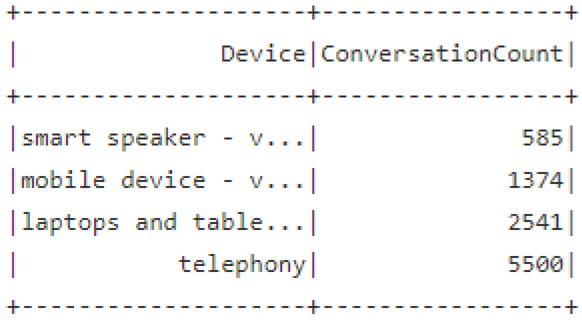
Now, let’s explore some relations between variables to produce more analytic insights.

Let’s start by aggregating the total conversation count by platform. To do this, create a new SQL script as above. Now, update the script to match the following:

```
SELECT [Device]
, COUNT(*) AS ConversationCount
FROM [default].[aiconvtable]
GROUP BY [Device]
ORDER BY [ConversationCount]
```

After setting “Connect to” to “ConvAiSQLPool,” press Run to execute the script.

This will return the total number of conversations across each platform. Here we can see that the fewest conversations occurred via smart speaker at 585 and the most occurred via telephony at 5,500:

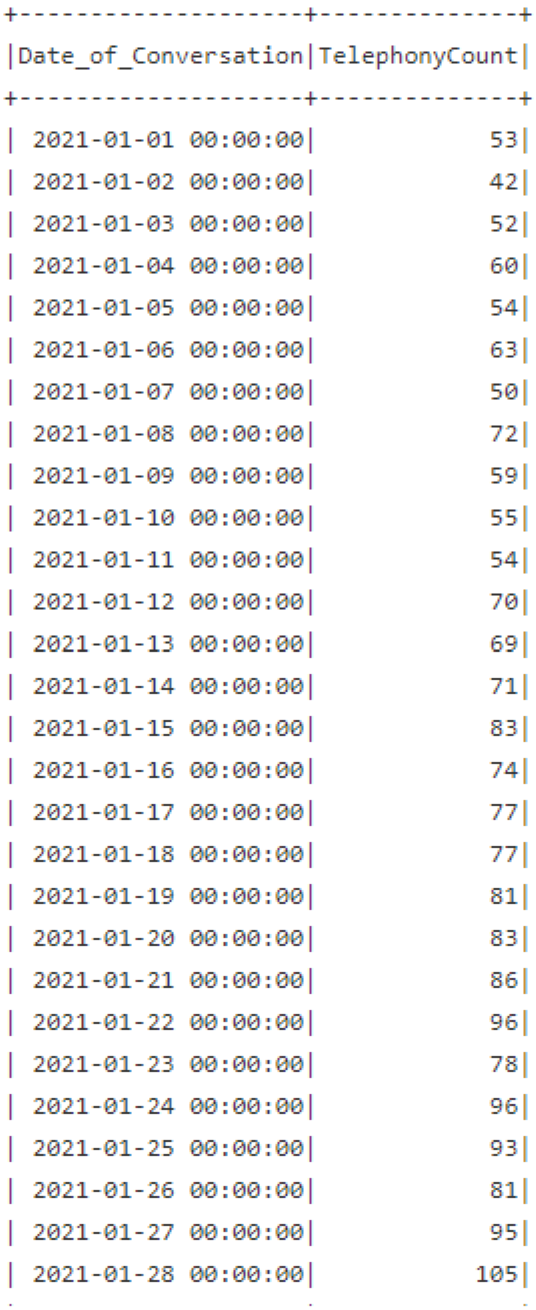


```
+--------------------+-----------------+
|              Device|ConversationCount|
+--------------------+-----------------+
|smart speaker - v...|              585|
|mobile device - v...|             1374|
|laptops and table...|             2541|
|           telephony|             5500|
+--------------------+-----------------+
```

But what if we want to be able to look at trends for a specific device over time? We can modify this query to the following:

```
SELECT [Date_of_Conversation]
, COUNT(*) AS TelephonyCount
WHERE [Device]=’telephony’
FROM [default].[aiconvtable]
GROUP BY [Date_of_Conversation]
ORDER BY [Date_of_Conversation]
```

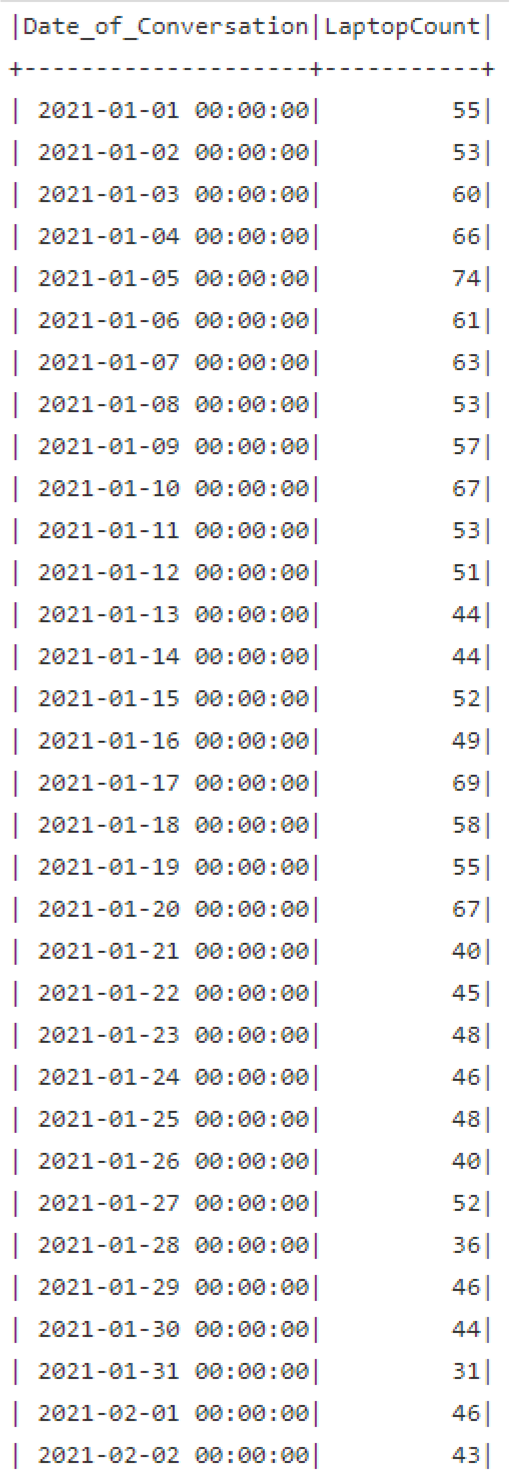
By looking at this data, we can see that the number of conversations occurring via telephony are generally increasing over time:



```
+-------------------+-------------+
|Date_of_Conversation|TelephonyCount|
+-------------------+-------------+
| 2021-01-01 00:00:00|           53|
| 2021-01-02 00:00:00|           42|
| 2021-01-03 00:00:00|           52|
| 2021-01-04 00:00:00|           60|
| 2021-01-05 00:00:00|           54|
| 2021-01-06 00:00:00|           63|
| 2021-01-07 00:00:00|           50|
| 2021-01-08 00:00:00|           72|
| 2021-01-09 00:00:00|           59|
| 2021-01-10 00:00:00|           55|
| 2021-01-11 00:00:00|           54|
| 2021-01-12 00:00:00|           70|
| 2021-01-13 00:00:00|           69|
| 2021-01-14 00:00:00|           71|
| 2021-01-15 00:00:00|           83|
| 2021-01-16 00:00:00|           74|
| 2021-01-17 00:00:00|           77|
| 2021-01-18 00:00:00|           77|
| 2021-01-19 00:00:00|           81|
| 2021-01-20 00:00:00|           83|
| 2021-01-21 00:00:00|           86|
| 2021-01-22 00:00:00|           96|
| 2021-01-23 00:00:00|           78|
| 2021-01-24 00:00:00|           96|
| 2021-01-25 00:00:00|           93|
| 2021-01-26 00:00:00|           81|
| 2021-01-27 00:00:00|           95|
| 2021-01-28 00:00:00|          105|
```

If we perform a similar analysis looking at the conversations occurring via laptops and tablets, we’ll notice an opposing trend, with conversation counts very gradually decreasing over time:

```
SELECT [Date_of_Conversation]
, COUNT(*) AS LaptopCount
WHERE [Device]=’laptops and tablets - screen’
FROM [default].[aiconvtable]
GROUP BY [Date_of_Conversation]
ORDER BY [Date_of_Conversation]
```

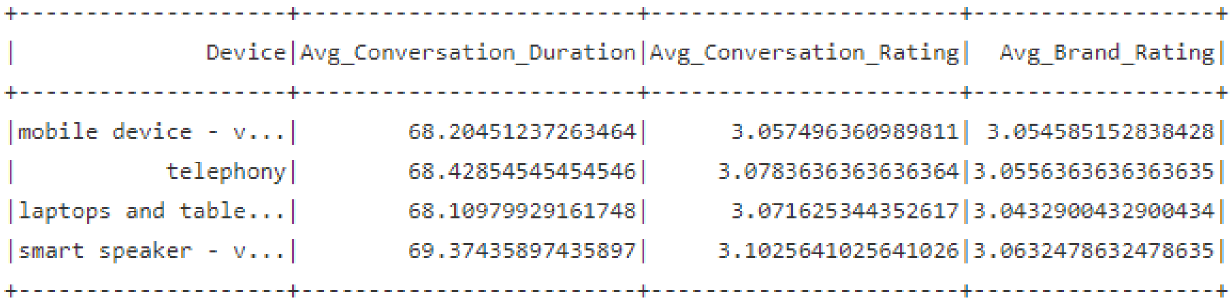


```
|Date_of_Conversation|LaptopCount|
+--------------------+-----------+
| 2021-01-01 00:00:00|         55|
| 2021-01-02 00:00:00|         53|
| 2021-01-03 00:00:00|         60|
| 2021-01-04 00:00:00|         66|
| 2021-01-05 00:00:00|         74|
| 2021-01-06 00:00:00|         61|
| 2021-01-07 00:00:00|         63|
| 2021-01-08 00:00:00|         53|
| 2021-01-09 00:00:00|         57|
| 2021-01-10 00:00:00|         67|
| 2021-01-11 00:00:00|         53|
| 2021-01-12 00:00:00|         51|
| 2021-01-13 00:00:00|         44|
| 2021-01-14 00:00:00|         44|
| 2021-01-15 00:00:00|         52|
| 2021-01-16 00:00:00|         49|
| 2021-01-17 00:00:00|         69|
| 2021-01-18 00:00:00|         58|
| 2021-01-19 00:00:00|         55|
| 2021-01-20 00:00:00|         67|
| 2021-01-21 00:00:00|         40|
| 2021-01-22 00:00:00|         45|
| 2021-01-23 00:00:00|         48|
| 2021-01-24 00:00:00|         46|
| 2021-01-25 00:00:00|         48|
| 2021-01-26 00:00:00|         40|
| 2021-01-27 00:00:00|         52|
| 2021-01-28 00:00:00|         36|
| 2021-01-29 00:00:00|         46|
| 2021-01-30 00:00:00|         44|
| 2021-01-31 00:00:00|         31|
| 2021-02-01 00:00:00|         46|
| 2021-02-02 00:00:00|         43|
```

Now let’s try looking at difference experience metrics by device type:

```
SELECT [Device]
, AVG([Conversation_Duration]) AS Avg_Conversation_Duration
, AVG([Rate_your_experience_-_Conversation]) As Avg_Conversation_Rating
, AVG([Rate_your_experience_-_Brand]) AS Avg_Brand_Rating
FROM [default].[aiconvtable]
GROUP BY [Device]
```

By executing this script, we can see that the differences between devices in these metrics are fairly slim:

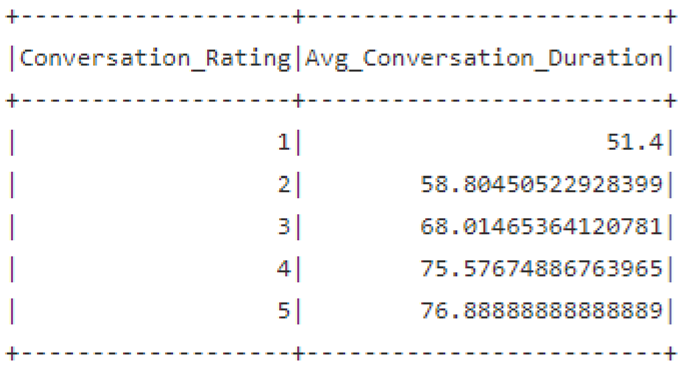


```
+------------------+----------------------+---------------------+-----------------+
|            Device|Avg_Conversation_Duration|Avg_Conversation_Rating|  Avg_Brand_Rating|
+------------------+----------------------+---------------------+-----------------+
|mobile device - v...|        68.20451237263464|      3.057496360989811| 3.054585152838428|
|         telephony|        68.42854545454546|     3.0783636363636364|3.0556363636363635|
|laptops and table...|        68.10979929161748|      3.071625344352617|3.0432900432900434|
|smart speaker - v...|        69.37435897435897|     3.1025641025641026|3.0632478632478635|
+------------------+----------------------+---------------------+-----------------+
```

Looking at that data, however, we can see that the conversation duration seems to be increasing along with the conversation rating. Let’s confirm if this is actually the case by zooming in on these two variables:

```
SELECT
[Rate_your_experience_-_Conversation]) As Conversation_Rating
, AVG([Conversation_Duration]) AS Avg_Conversation_Duration
FROM [default].[aiconvtable]
ORDER BY [Avg_Conversation_Duration]
```

We can see from the output that the average conversation duration is in fact longer for conversations with higher ratings:



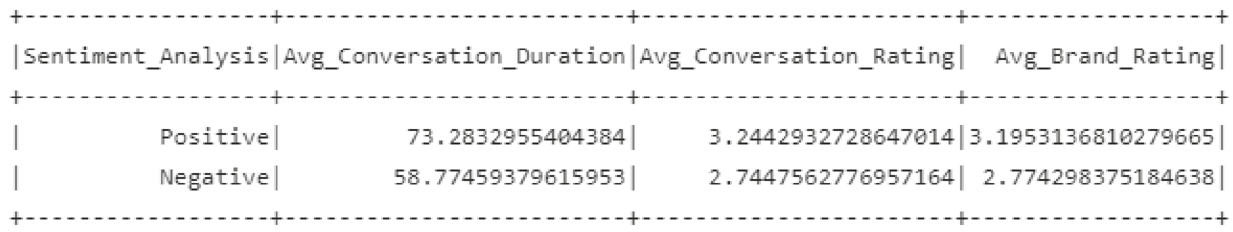
```
+-------------------+-------------------------+
|Conversation_Rating|Avg_Conversation_Duration|
+-------------------+-------------------------+
|                  1|                     51.4|
|                  2|        58.80450522928399|
|                  3|        68.01465364120781|
|                  4|        75.57674886763965|
|                  5|        76.88888888888889|
+-------------------+-------------------------+
```

We can also see a similar trend when grouping conversations based on sentiment analysis:

```
SELECT
[Sentiment_Analysis]
, AVG([Conversation_Duration]) AS Avg_Conversation_Duration
, AVG([Rate_your_experience_-_Conversation]) As Avg_Conversation_Rating
, AVG([Rate_your_experience_-_Brand]) AS Avg_Brand_Rating
```

FROM [default].[aiconvtable]
GROUP BY [Sentiment_Analysis]

Conversations with positive sentiment analysis tend to have longer conversation durations, higher average conversation ratings, and higher average brand ratings:



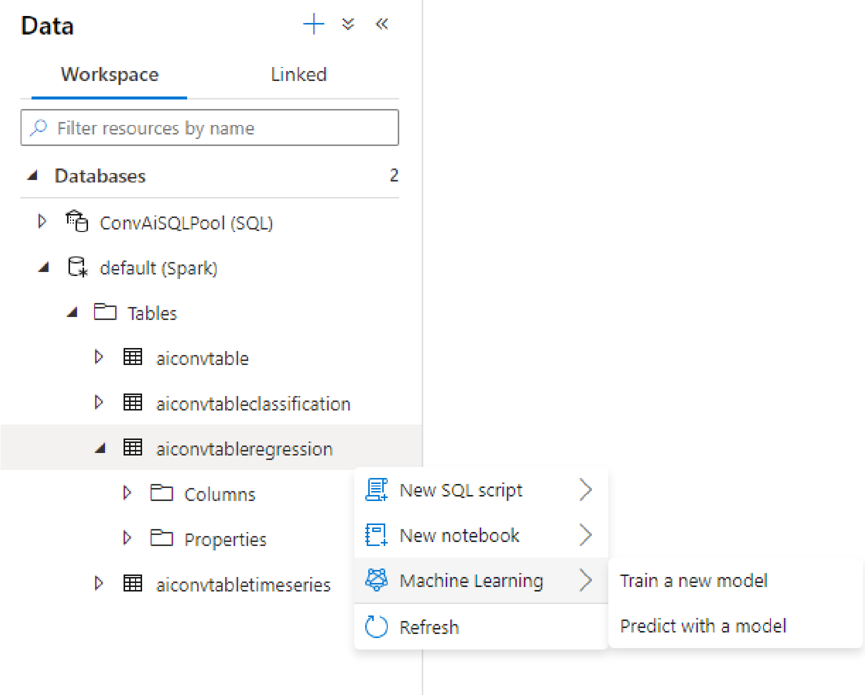
```
+------------------+-------------------------+-----------------------+------------------+
|Sentiment_Analysis|Avg_Conversation_Duration|Avg_Conversation_Rating|  Avg_Brand_Rating|
+------------------+-------------------------+-----------------------+------------------+
|          Positive|         73.2832955404384|     3.2442932728647014|3.1953136810279665|
|          Negative|        58.77459379615953|     2.7447562776957164| 2.774298375184638|
+------------------+-------------------------+-----------------------+------------------+
```

# Launch a Regression Model in Azure Synapse

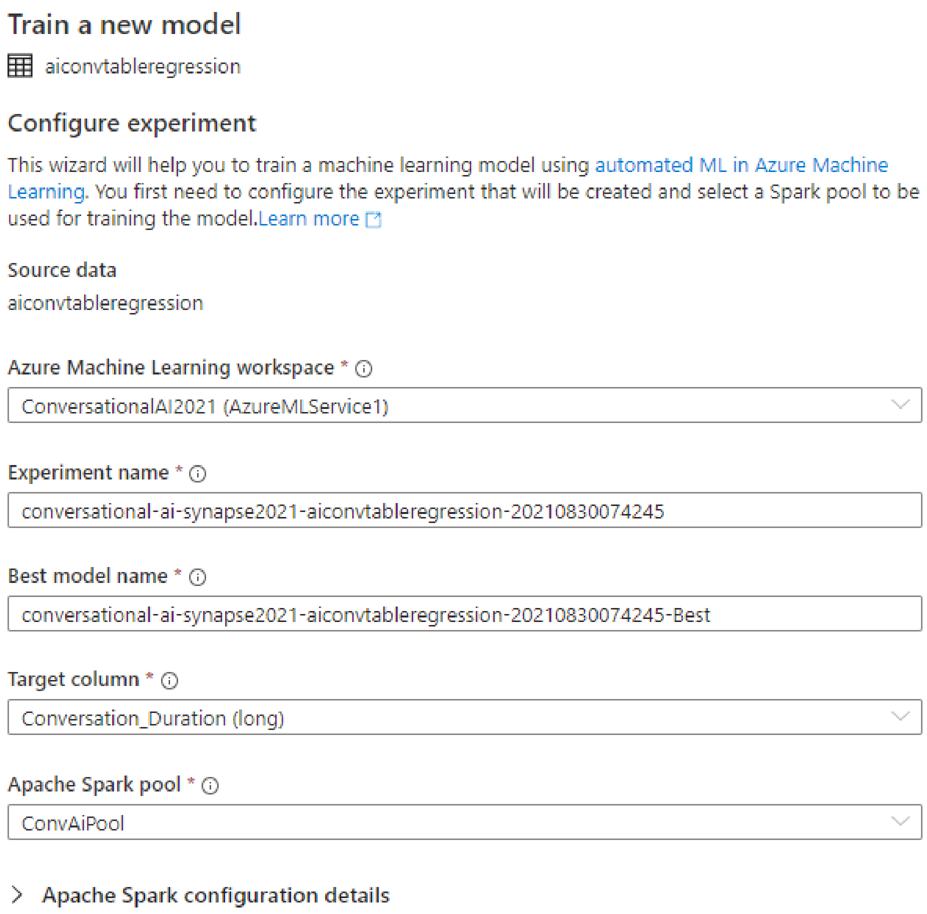
## Train the Regression Model

Now that we've explored our data, we're ready to create an ML model from within Azure Synapse. In the Azure Synapse workspace, navigate to the "Data" option in the left pane. Under Databases, select "default (Spark)" and then "Tables."

1. Select the data that will be used to train the regression model, in our case "aiconvtableregression."
2. Select the 3-dot menu next this table
3. Select "Machine Learning"
4. Select "Train a new model"



Set the target column to conversation duration, the factor that the model will predict:



## Train a new model

aiconvtableregression

### Configure experiment

This wizard will help you to train a machine learning model using automated ML in Azure Machine Learning. You first need to configure the experiment that will be created and select a Spark pool to be used for training the model.Learn more

Source data
aiconvtableregression

Azure Machine Learning workspace *
ConversationalAI2021 (AzureMLService1)

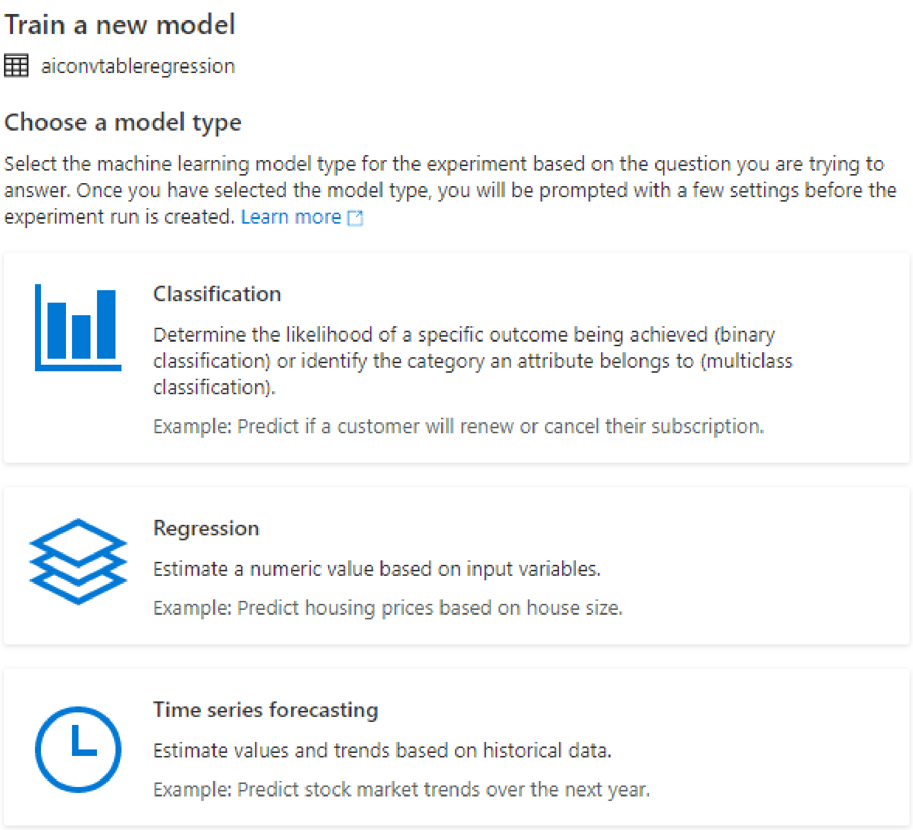
Experiment name *
conversational-ai-synapse2021-aiconvtableregression-20210830074245

Best model name *
conversational-ai-synapse2021-aiconvtableregression-20210830074245-Best

Target column *
Conversation_Duration (long)

Apache Spark pool *
ConvAiPool

Click next. Select “regression” as the model type:



Train a new model

aiconvtableregression

Choose a model type

Select the machine learning model type for the experiment based on the question you are trying to answer. Once you have selected the model type, you will be prompted with a few settings before the experiment run is created. Learn more

Classification

Determine the likelihood of a specific outcome being achieved (binary classification) or identify the category an attribute belongs to (multiclass classification).

Example: Predict if a customer will renew or cancel their subscription.

Regression

Estimate a numeric value based on input variables.

Example: Predict housing prices based on house size.

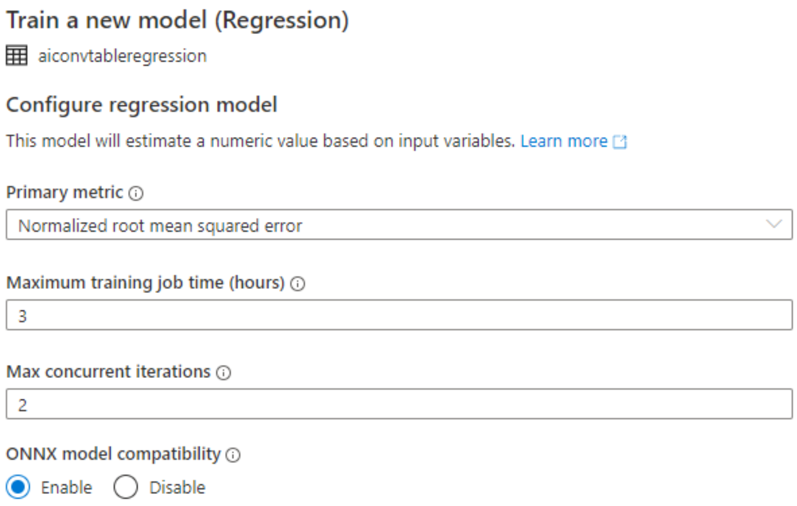
Time series forecasting

Estimate values and trends based on historical data.

Example: Predict stock market trends over the next year.

Click Continue. On the next screen, set the primary metric to “normalized root squared error” and set ONNX model compatibility to enabled. This metric is preferred over Spearman correlation because we’re looking to predict accurate conversation durations rather than the order of their durations. We can leave the other metrics the same.

Your settings should look like this:



Train a new model (Regression)

aiconvtableregression

Configure regression model

This model will estimate a numeric value based on input variables. Learn more

Primary metric: Normalized root mean squared error

Maximum training job time (hours): 3

Max concurrent iterations: 2

ONNX model compatibility

● Enable ○ Disable

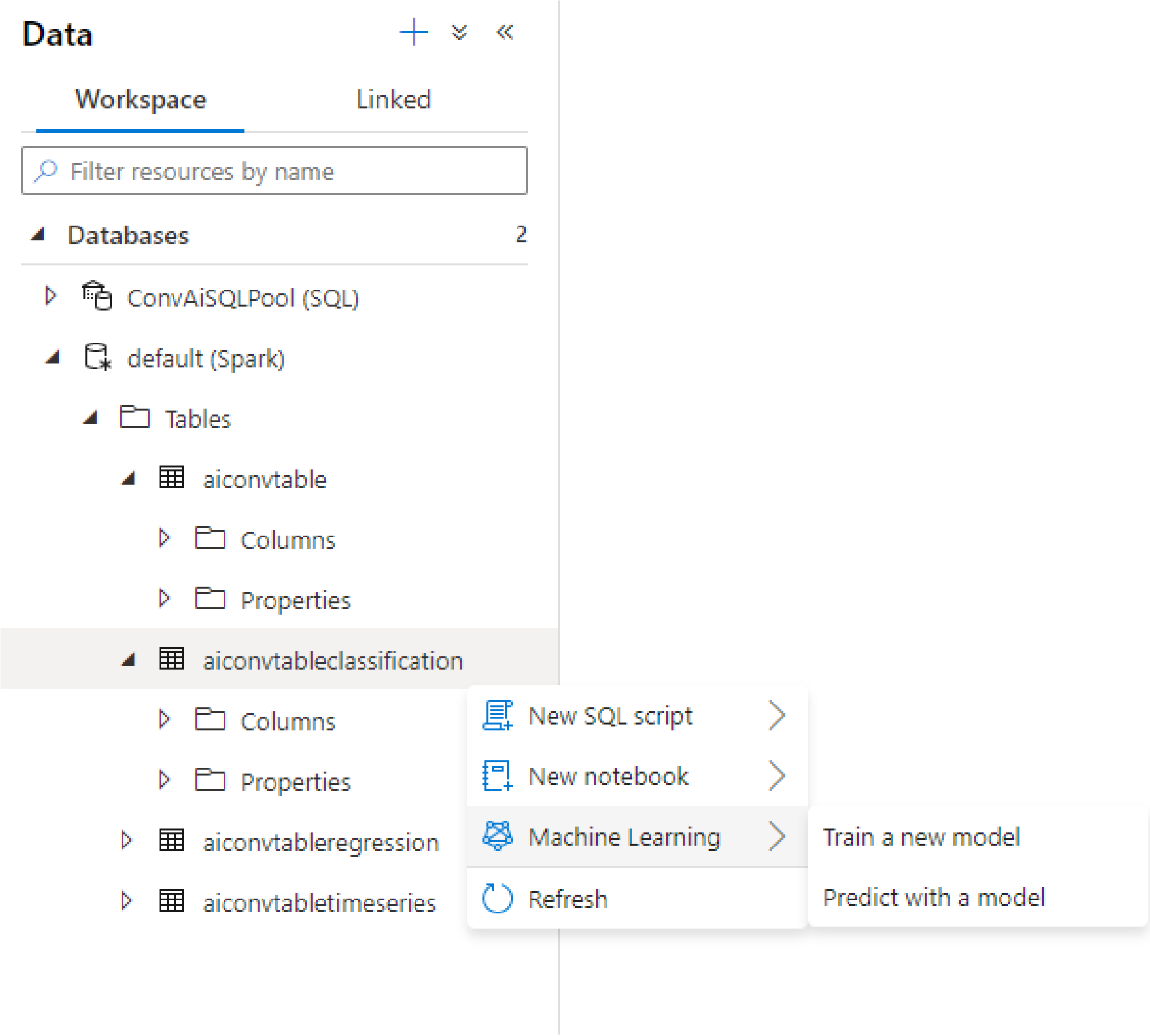
Click “Create Run” to start the training process. This may take a few hours to run. Once the model is complete, the results can be explored from within the ML workspace.

## Launch a Classification Model in Azure Synapse

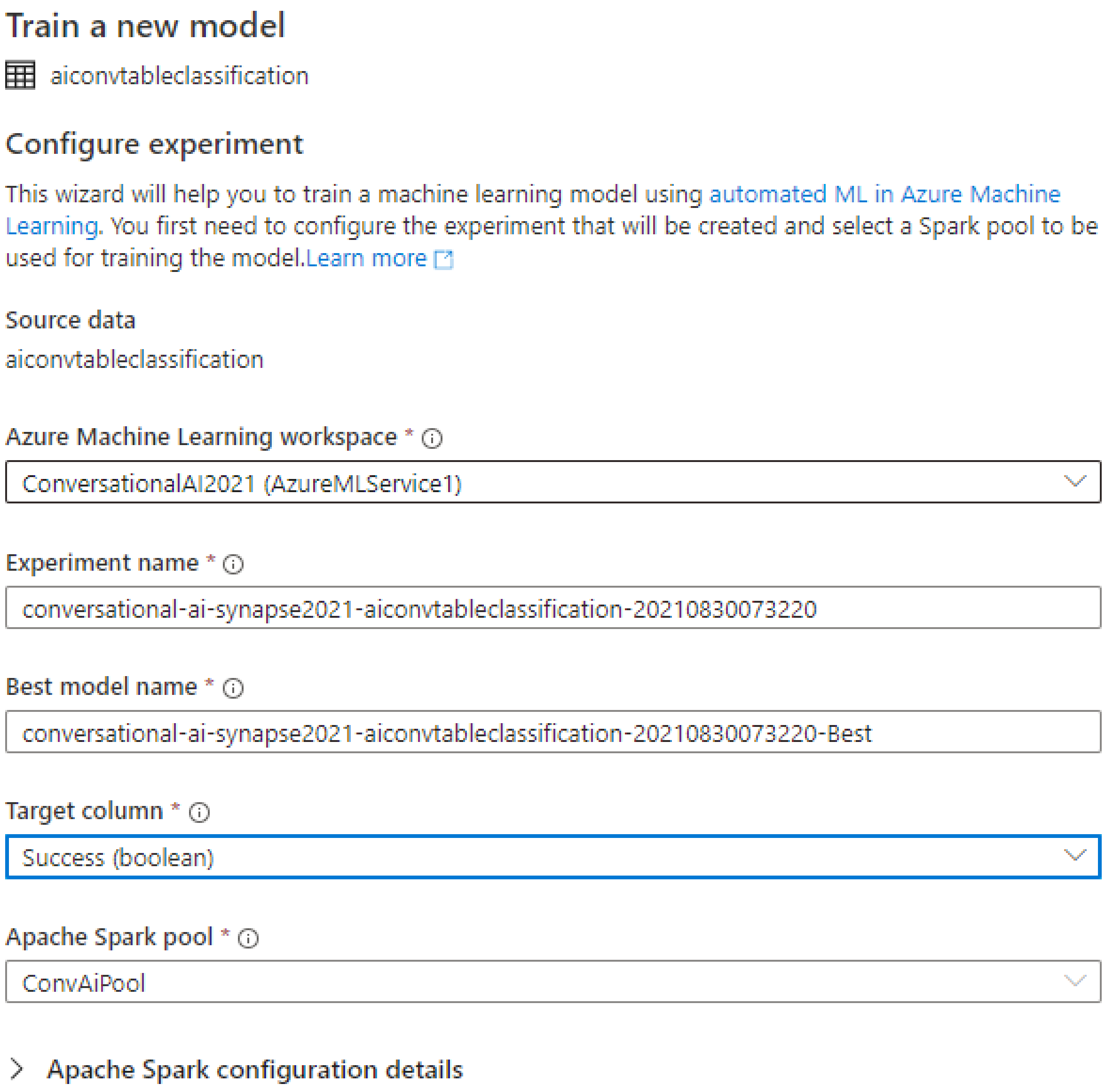
### Train the Classification Model

In the Azure Synapse workspace, navigate to the “Data” option in the left pane. Under Databases, select “default (Spark)” and then “Tables.”

1. Select the data that will be used to train the classification model, in our case “aiconvtableclassification.”
2. Select the 3-dot menu next this table
3. Select “Machine Learning”
4. Select “Train a new model”



Set the target column to success, the factor that the model should predict:



## Train a new model

aiconvtableclassification

### Configure experiment

This wizard will help you to train a machine learning model using automated ML in Azure Machine Learning. You first need to configure the experiment that will be created and select a Spark pool to be used for training the model.Learn more

Source data

aiconvtableclassification

Azure Machine Learning workspace *

ConversationalAI2021 (AzureMLService1)

Experiment name *

conversational-ai-synapse2021-aiconvtableclassification-20210830073220

Best model name *

conversational-ai-synapse2021-aiconvtableclassification-20210830073220-Best

Target column *

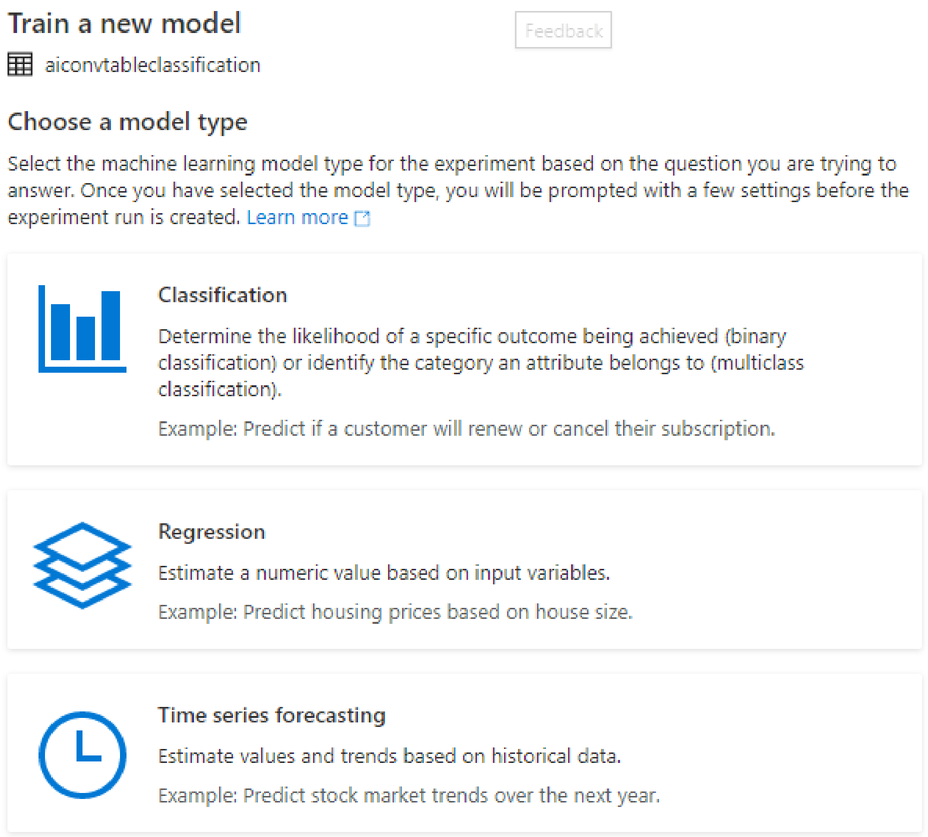
Success (boolean)

Apache Spark pool *

ConvAiPool

> Apache Spark configuration details

Click next. Select “classification” as the model type:



Train a new model

aiconvtableclassification

Choose a model type

Select the machine learning model type for the experiment based on the question you are trying to answer. Once you have selected the model type, you will be prompted with a few settings before the experiment run is created. Learn more

Classification

Determine the likelihood of a specific outcome being achieved (binary classification) or identify the category an attribute belongs to (multiclass classification).

Example: Predict if a customer will renew or cancel their subscription.

Regression

Estimate a numeric value based on input variables.

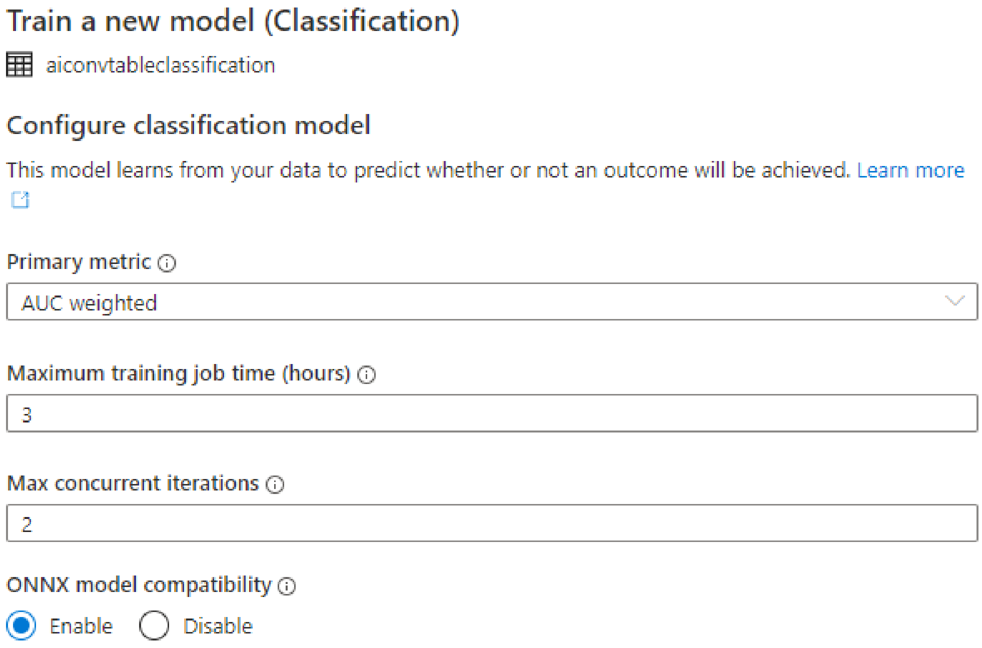
Example: Predict housing prices based on house size.

Time series forecasting

Estimate values and trends based on historical data.

Example: Predict stock market trends over the next year.

Click continue. On the next screen, change the primary metric to AUC weighted and set ONNX model compatibility to enabled. We can leave the other options alone. Your settings should look like this:



Train a new model (Classification)

aiconvtableclassification

Configure classification model

This model learns from your data to predict whether or not an outcome will be achieved. Learn more

Primary metric ⓘ

AUC weighted

Maximum training job time (hours) ⓘ

3

Max concurrent iterations ⓘ

2

ONNX model compatibility ⓘ

● Enable ○ Disable

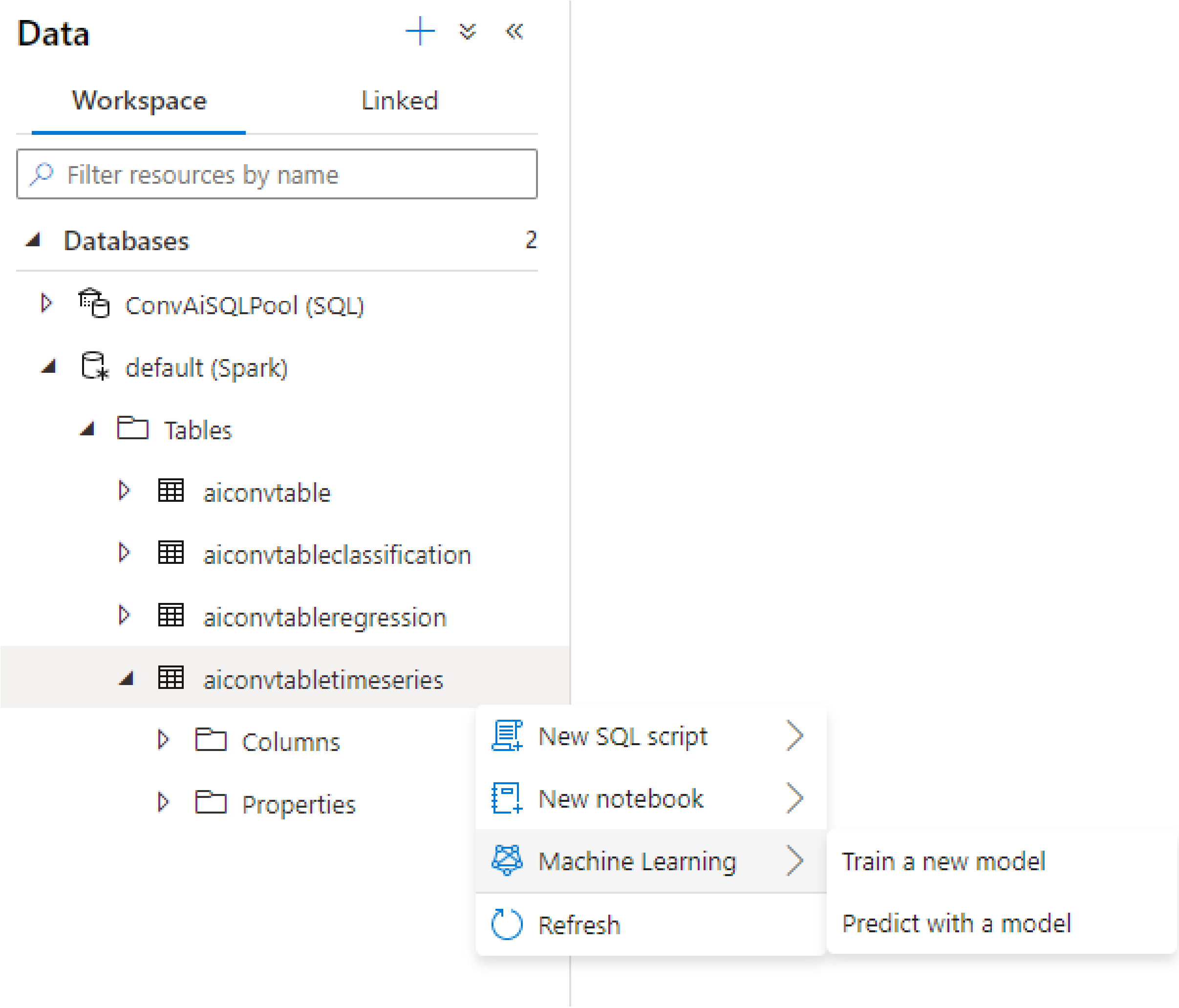
Click “Create Run” to start the training process. This may take a few hours to run. Once the model is complete, the results can be explored from within the ML workspace.

## Launch a Time Series Model in Azure Synapse

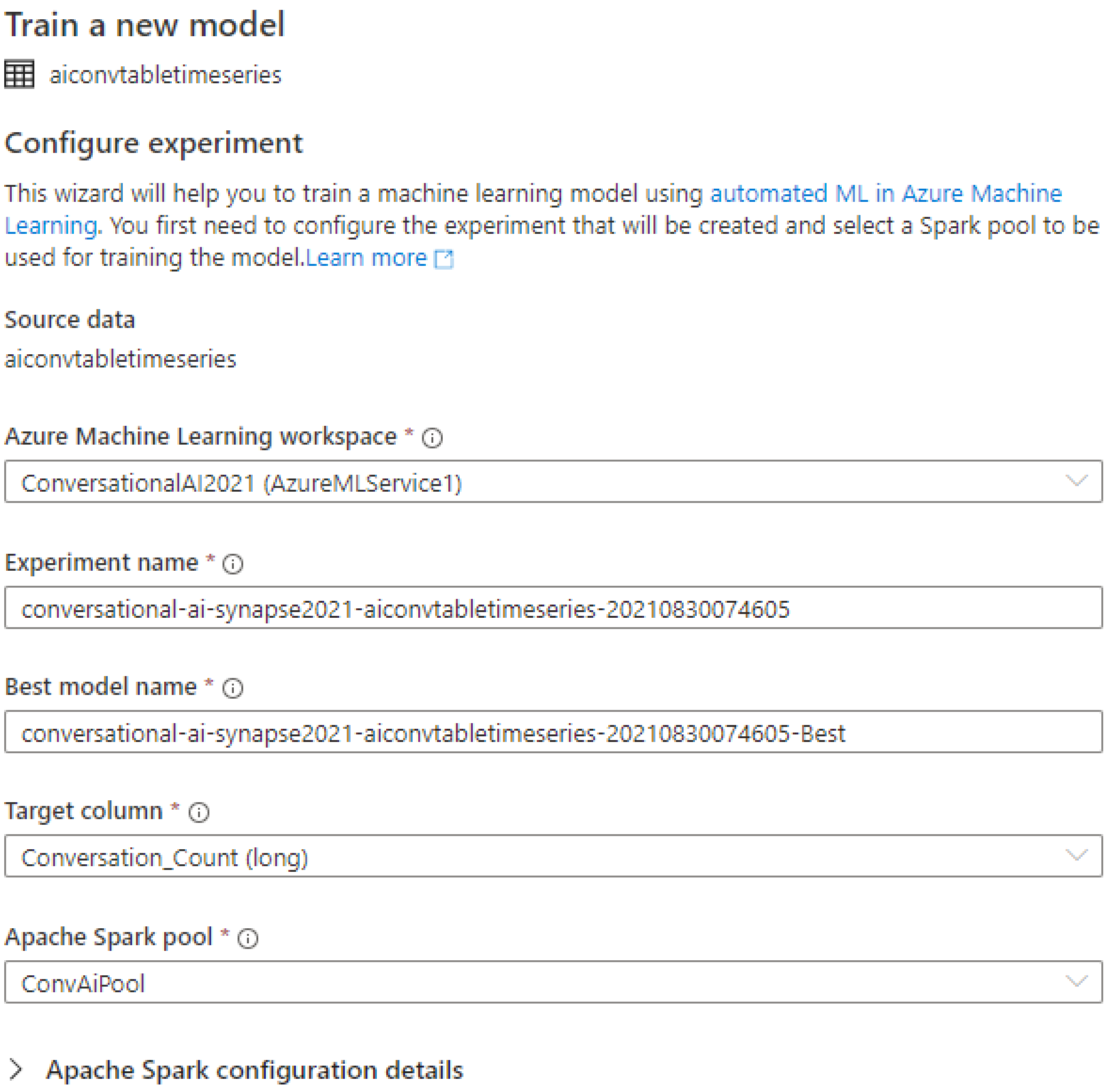
### Train the Time Series Model

In the Azure Synapse workspace, navigate to the “Data” option in the left pane. Under Databases, select “default (Spark)” and then “Tables.”

1. Select the data that will be used to train the regression model, in our case “aiconvtabletimeseries”
2. Select the 3-dot menu next this table
3. Select “Machine Learning”
4. Select “Train a new model”



Set the target column to success, the factor that the model should predict:



## Train a new model

aiconvtabletimeseries

### Configure experiment

This wizard will help you to train a machine learning model using automated ML in Azure Machine Learning. You first need to configure the experiment that will be created and select a Spark pool to be used for training the model.Learn more

Source data

aiconvtabletimeseries

Azure Machine Learning workspace *

ConversationalAI2021 (AzureMLService1)

Experiment name *

conversational-ai-synapse2021-aiconvtabletimeseries-20210830074605

Best model name *

conversational-ai-synapse2021-aiconvtabletimeseries-20210830074605-Best

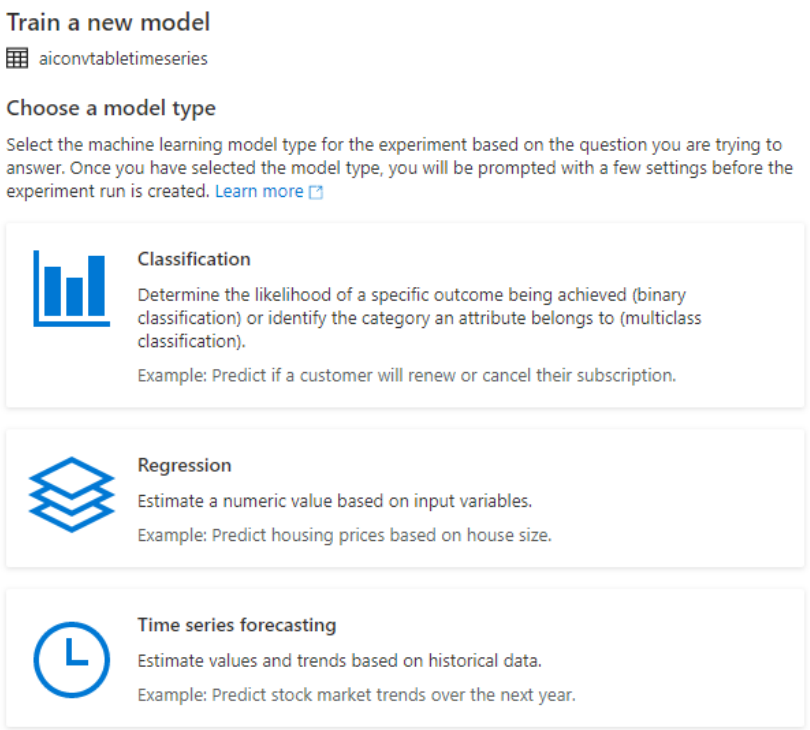
Target column *

Conversation_Count (long)

Apache Spark pool *

ConvAiPool

Click next. Select “time series forecasting” as the model type:



Train a new model

aiconvtabletimeseries

Choose a model type

Select the machine learning model type for the experiment based on the question you are trying to answer. Once you have selected the model type, you will be prompted with a few settings before the experiment run is created. Learn more

Classification

Determine the likelihood of a specific outcome being achieved (binary classification) or identify the category an attribute belongs to (multiclass classification).

Example: Predict if a customer will renew or cancel their subscription.

Regression

Estimate a numeric value based on input variables.

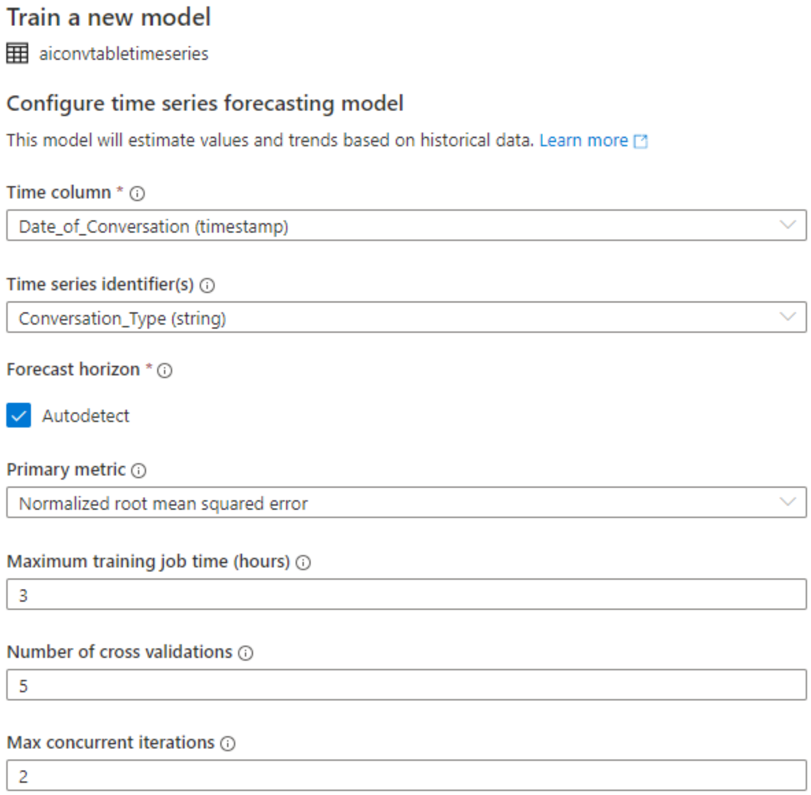
Example: Predict housing prices based on house size.

Time series forecasting

Estimate values and trends based on historical data.

Example: Predict stock market trends over the next year.

On the next screen, set the "time column" to the date of conversation variable. Under time series identifier(s) check "conversation_type". Your settings should look like this:



Train a new model

aiconvtabletimeseries

Configure time series forecasting model

This model will estimate values and trends based on historical data. Learn more

Time column *

Date_of_Conversation (timestamp)

Time series identifier(s)

Conversation_Type (string)

Forecast horizon *

☑ Autodetect

Primary metric

Normalized root mean squared error

Maximum training job time (hours)

3

Number of cross validations

5

Max concurrent iterations

2

Click “Create Run” to start the training process. This may take a few hours to run. Once the model is complete, the results can be explored from within the ML workspace.