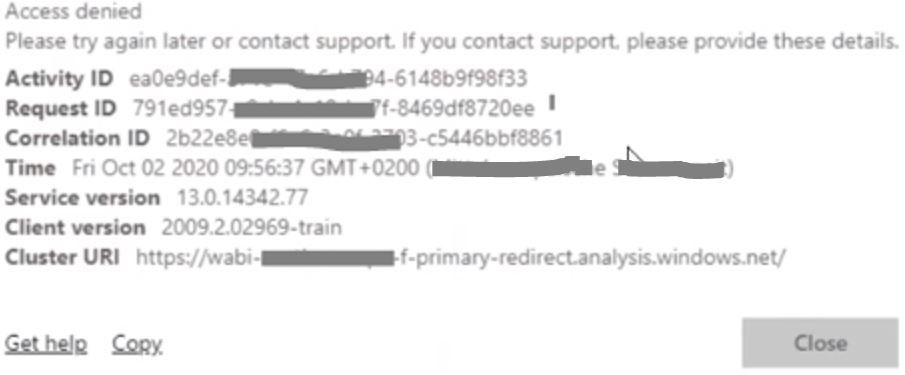
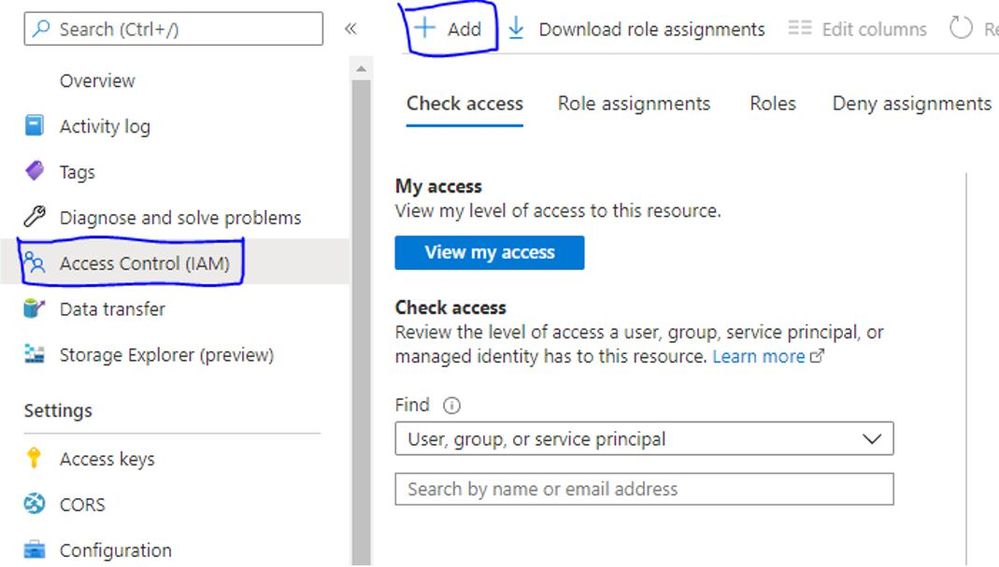
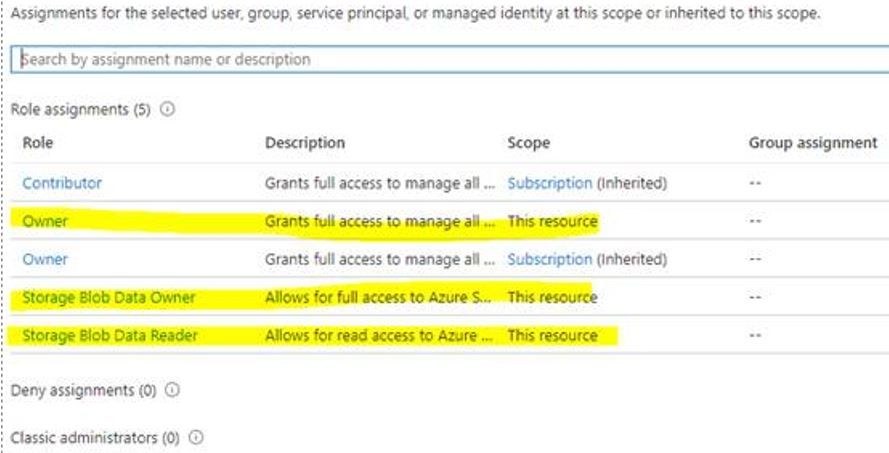
DEMO 1

1. Creare un workspace **Dataflow**.
2. Mostrare che si può assegnare un **Dataflow** ad una versione **Azure Data Lake** Personale e non usare quella di default
   1. Molto probabilmente darà errore quello che si vede sotto

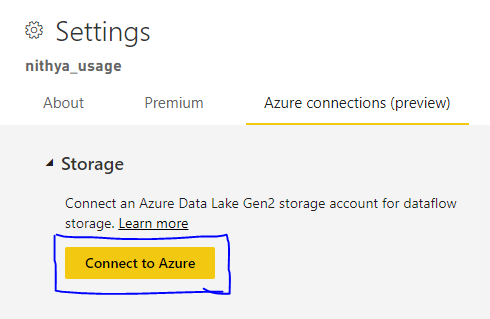


* 1. Se da errore bisogna abilitare i ruoli





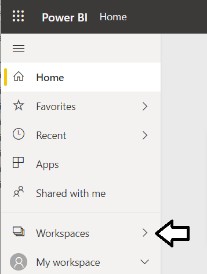
* 1. Poi si può procedere a collegare il workspace al data lake



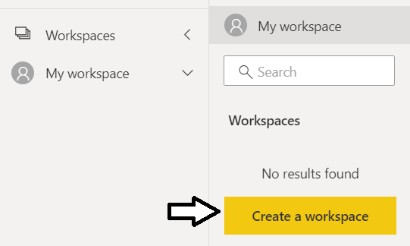
Demo 2

# Exercise 1: Use Power Query to Create a New Dataflow

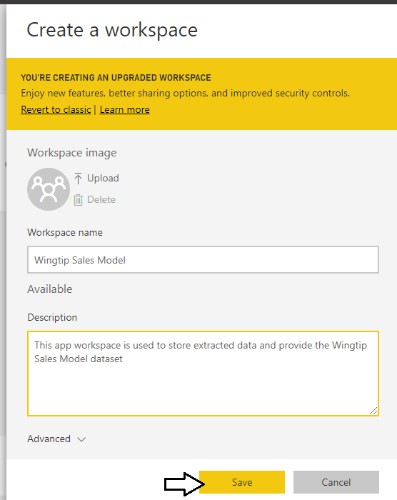
1. Log into the Power BI Service with your new organizational account.
   1. Navigate the Power BI portal at [https://app.powerbi.com](https://app.powerbi.com/) and if prompted, log in using your organizational account.
2. Create a new app workspace named **Wingtip Sales Model**.
   1. Click the **Workspace** flyout menu in the left navigation.



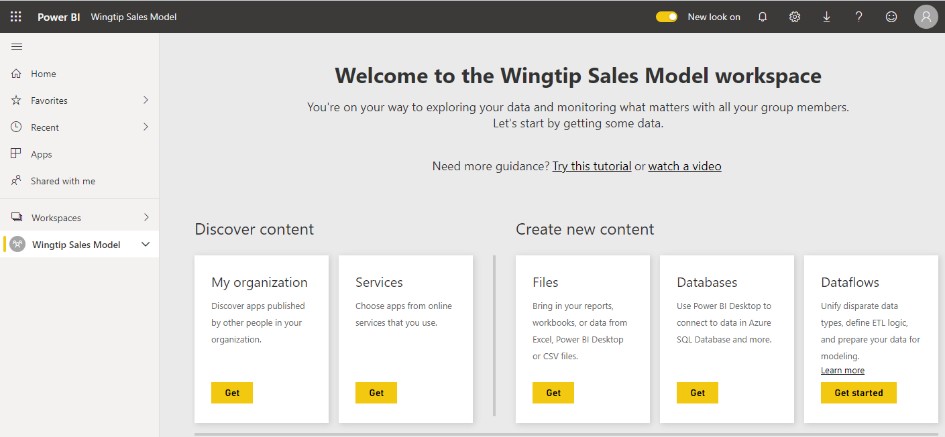
* 1. Click the **Create app workspace** button to display the **Create an app workspace** dialog.



* 1. In the **Create an app workspace** pane, enter a workspace name of **Wingtip Sales Model**.
  2. Click the **Save** button to create the new app workspace named **Wingtip Sales Model**.

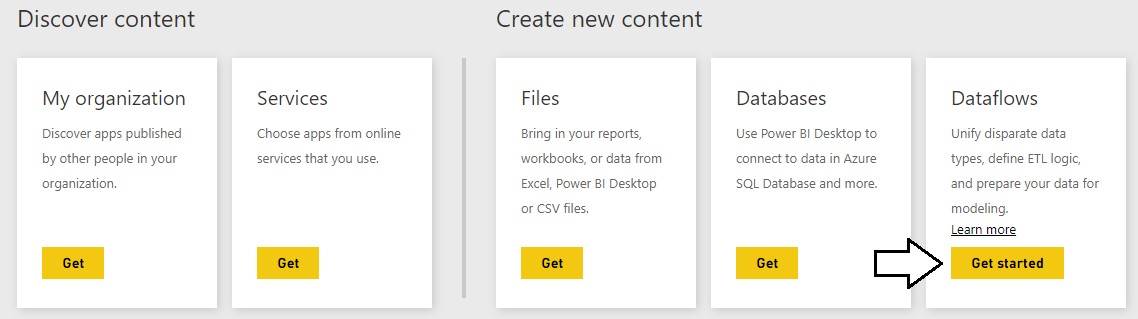


* 1. When you click **Save**, the Power BI service should create the new app workspace and then switch your current Power BI session to be running within the context of the new **Wingtip Sales Model** workspace.

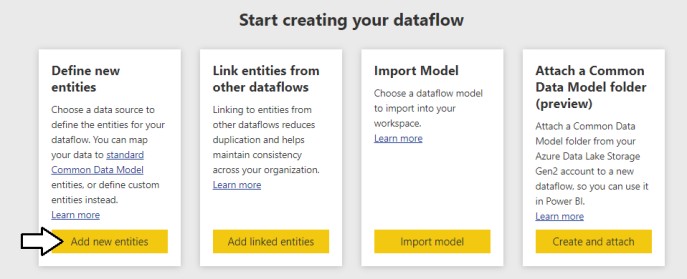


You should now see the welcome page for the **Wingtip Sales Model** workspace. The welcome page is shown by default when an app workspace is empty and contains no content.

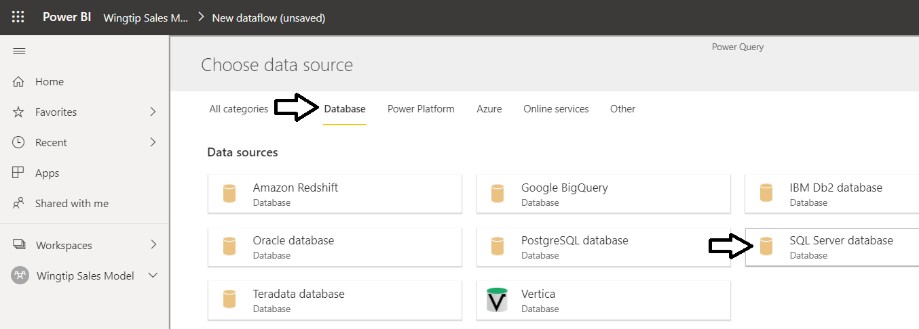
1. Create a new dataflow.
   1. Click the **Get started** button in the **Dataflows** section.



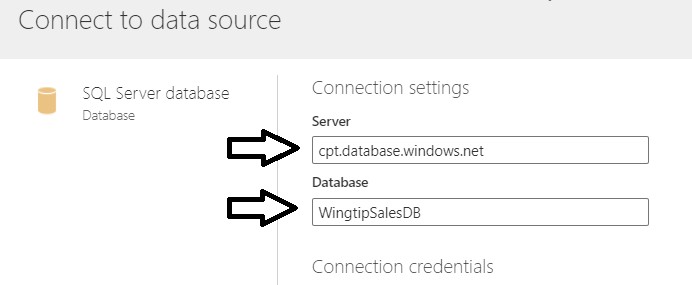
* 1. Click the **Add new entities** button in the **Define new entities** section.



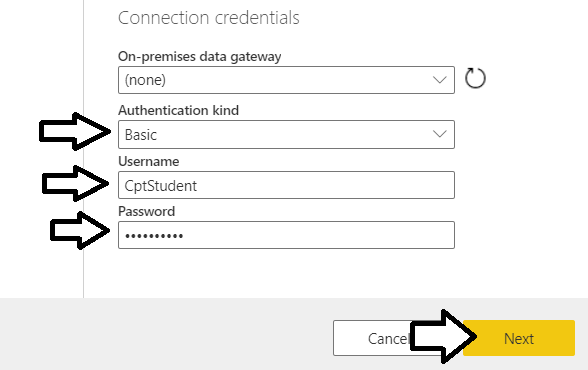
* 1. On the **Choose data source** page, select the **Database** tab and then select **SQL Server database**.



* 1. Enter a **Server** value of **afterhourdemo-ondemand.sql.azuresynapse.net**
  2. Enter a **Database** value of **Demo**.

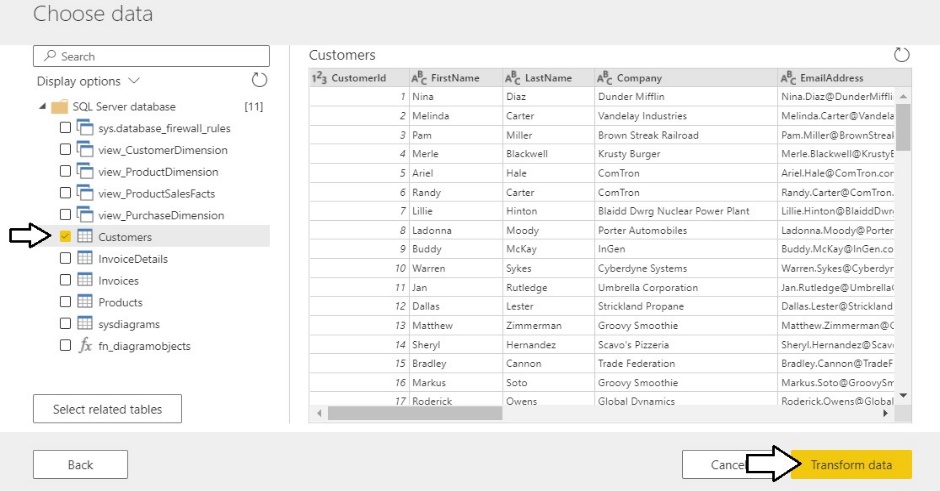


* 1. Move down the **Connection credentials** section.
  2. Set the **Authentication kind** setting to **Basic**.
  3. Enter a **Username** of **sqladminuser**.
  4. Enter a **Password** of **Pa$$w0rd**
  5. Click the **Next** button to continue.



Once you establish a connection, you should be promoted with the **Choose data** screen to select one or more tables.

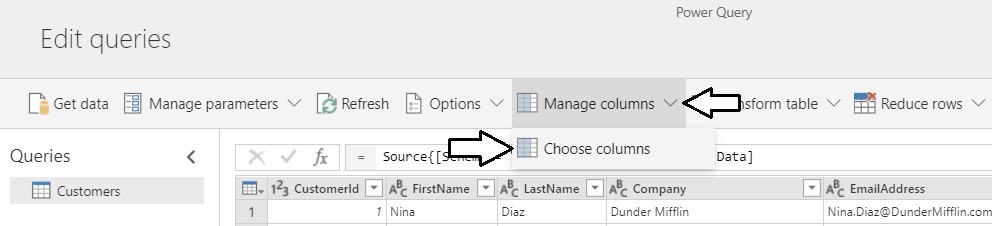
1. Select the **Customers** table
   1. In the **Choose data** section, select the **Customers** table and then click **Transform data**.



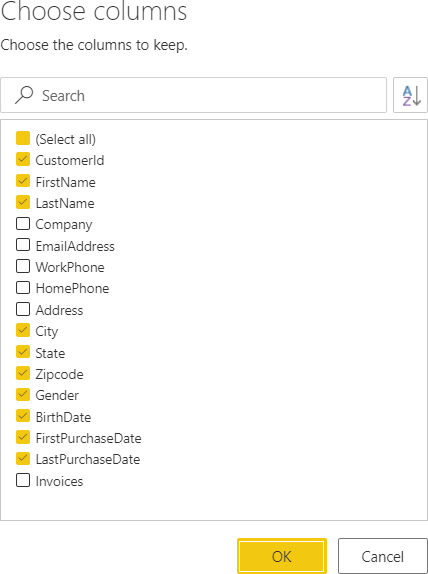
* 1. You should now see the **Edit queries** screen displaying query results and the M formula bar for the current step.



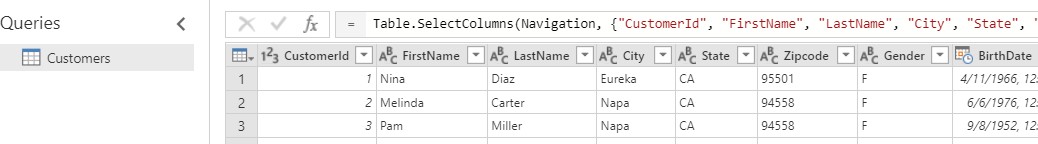
1. Use Power Query to clean and transform the data from the **Customers** table.
   1. Drop down the **Manage columns** menu button and select the **Choose columns** command.



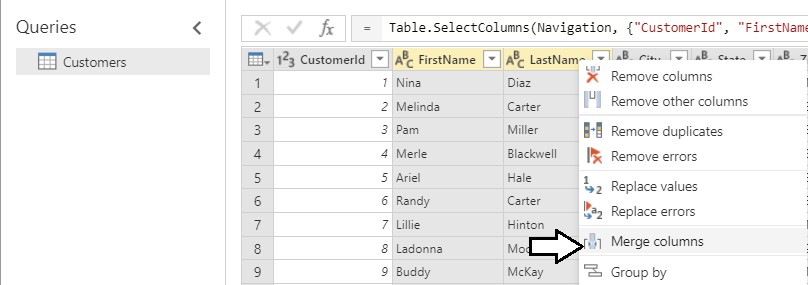
* 1. In the **Choose columns** dialog, begin by clicking on the **(Select all)** checkbox at the top to unselect all column. Next, select the checkboxes for the following columns.
     1. **CustomerId**
     2. **FirstName**
     3. **LastName**
     4. **City**
     5. **State**
     6. **Zipcode**
     7. **Gender**
     8. **BirthDate**
     9. **FirstPurchaseDate**
     10. **LastPurchaseDate**
  2. Once you have the columns selected as shown in the following screenshot, click **OK** to close the **Choose columns** dialog.



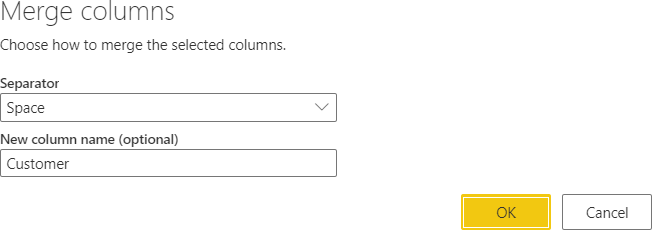
* 1. You should be able to verify that the Power Query editor now only shows the columns that you selected.



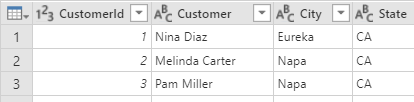
1. In this step you will merge the **FirstName** column and the **LastName** column together into a single column named **Customer**.
   1. Select the **FirstName** column by clicking on its column header.
   2. Next, hold down the **SHIFT** key and select the **LastName** column by clicking on its column header.
   3. Right-click on the selected columns and click the **Merge columns** menu command.



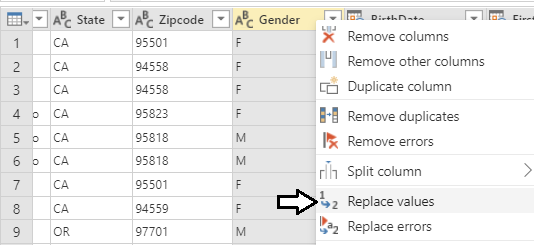
* 1. In the **Merge Column** dialog, drop down the **Separator** control and select a value of **Space**.
  2. Add a **New column name** value of **Customer** and click the **OK** button to modify the underlying query with your changes.



* 1. You should now be able to see that the **FirstName** column and the **LastName** column have been replaced with a single merged column named **Customer**.

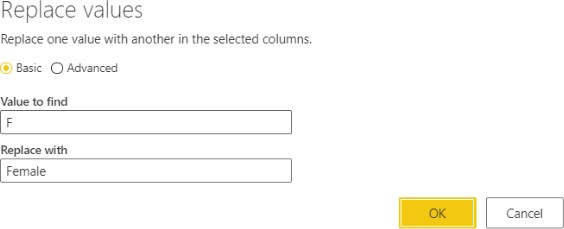


1. Modify the query so that the **Gender** column returns values of **Male** and **Female** instead of **M** and **F**.
   1. Locate the **Gender** column in the **Customers** table.
   2. Right-click the header for the **Gender** column and select the **Replace values** command to display the **Replace values** dialog.

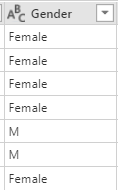


* 1. In the **Replace values** dialog, enter a value of **F** in the **Value to find** textbox and enter a value of **Female** in the **Replace with**

textbox. Click to **OK** button add your changes to the underlying query.

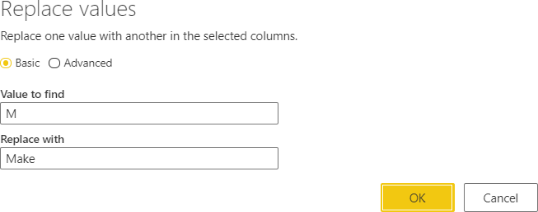


* 1. You should be able to see that all values of **F** in the **Gender** column have been replaced with a value of **Female**.



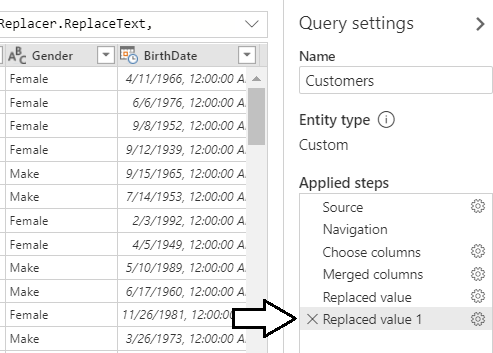
* 1. Right-click the header for the **Gender** column and select the **Replace values** command a second time.
  2. In the **Replace values** dialog, enter a value of **M** in the **Value to find** textbox and enter a value of **Male** in the **Replace with**

textbox. Click to **OK** button add your changes to the underlying query.

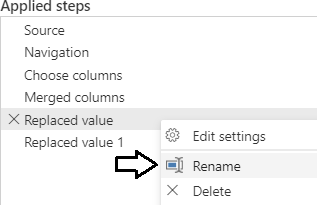


* 1. You should be able to confirm that all values in the **Gender** column have been replaced with a value of either **Male** or **Female**.

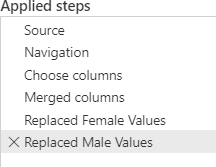
1. Change the name of query steps.
   1. Inspect the **Applied Steps** list in the **Query settings** pane. You should be able to see that there are two steps at the end that have been given the generic names of **Replaced value** and **Replaced value 1**.

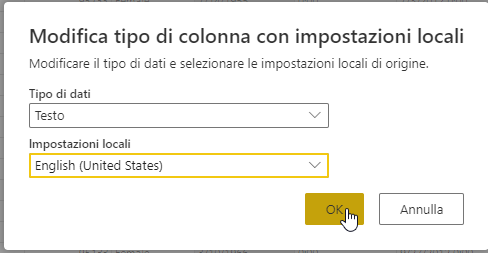


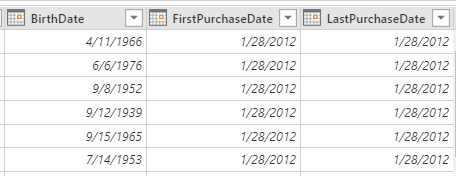
* 1. Rename the **Replaced Values** step by right-clicking it and clicking the **Rename** command to place the step name in edit mode. Modify the name of this step to **Replace Female Values**.



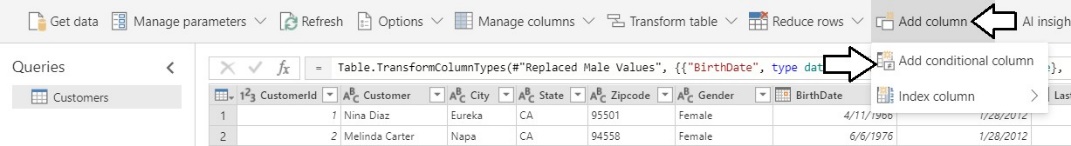
* 1. Using the same technique, rename the **Replaced Value 1** step to **Replaced Male Values**.



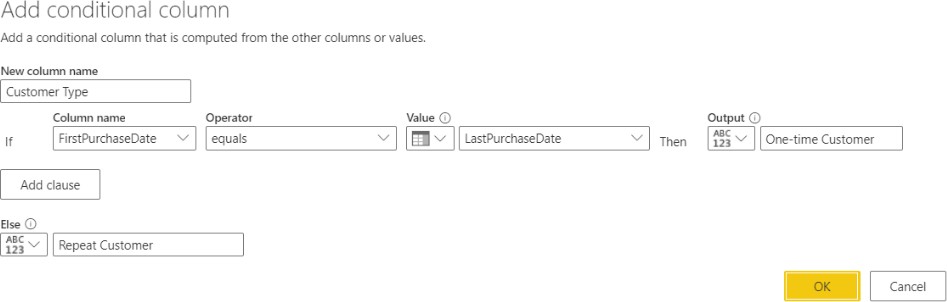
1. Change the column type of **BirthDate**, **FirstPurchasedDate** and **LastPurchasedDate** from **Date/Time** to **Date**.
   1. Dividi la colonna birthdate in base al delimitatore spazio
   2. Elimina altra colonna
   3. Use the column type drop down on the left-hand side of the **BirthDate** column to configure the column using the **Date** type with location
   4. Use the datatype drop down menu of the **FirstPurchaseDate** column to configure the column using the **Date** type.
   5. Use the datatype drop down menu of the **LastPurchaseDate** column to configure the column using the **Date** type.
   6. You should see that the three columns now show values with a date but without a time.



1. Add a new conditional column named **Customer Type** to indicates whether the customer is a repeat customer or not.
   1. Drop down the **Add column** menu button and select the **Add conditional column** command.



* 1. In the **Add conditional column** dialog, enter a **New column name** value of **Customer Type**.
  2. Configure a rule to return a string value of “One-time Customer” if **FirstPurchaseDate** equals **LastPurchaseDate**.
  3. For the **Else** evaluation, return a string value of “Repeat Customer”.
  4. When the **Add conditional column** dialog matches the screenshot below, click the **OK** button to add the new column.



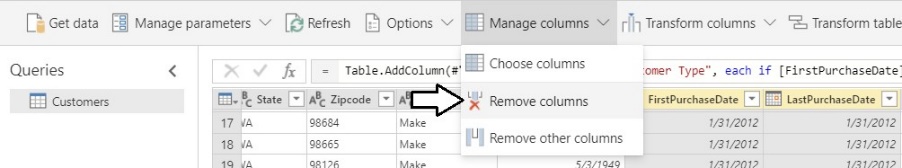
* 1. You should be able to verify that the new **Customer Type** column has a value of **Repeat Customer** when the current customer has a **FirstPurchaseDate** column value that is not equal to the **LastPurchaseDate** column value. When these column values are equal, the **Customer Type** column has a value of **One-time Customer**.

Now, that you are done using the **FirstPurchaseDate** column and the **LastPurchaseDate** column to calculate the value of the

**Customer Type** column, you can delete them because they are no longer needed.

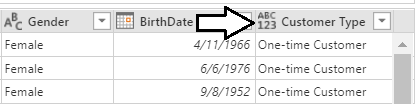


1. Remove the **FirstPurchaseDate** column and the **LastPurchaseDate** column.
   1. Select the **FirstPurchaseDate** column by clicking its column header.
   2. Hold down the **SHIFT** key and click the column header for **LastPurchaseDate** so that both columns are selected.
   3. Right click the one of the selected columns and click the **Remove Columns**.

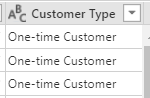


* 1. You should be able to confirm that the **FirstPurchaseDate** column and the **LastPurchaseDate** columns have been removed from the query results. However, the **Customer Type** column is still there.

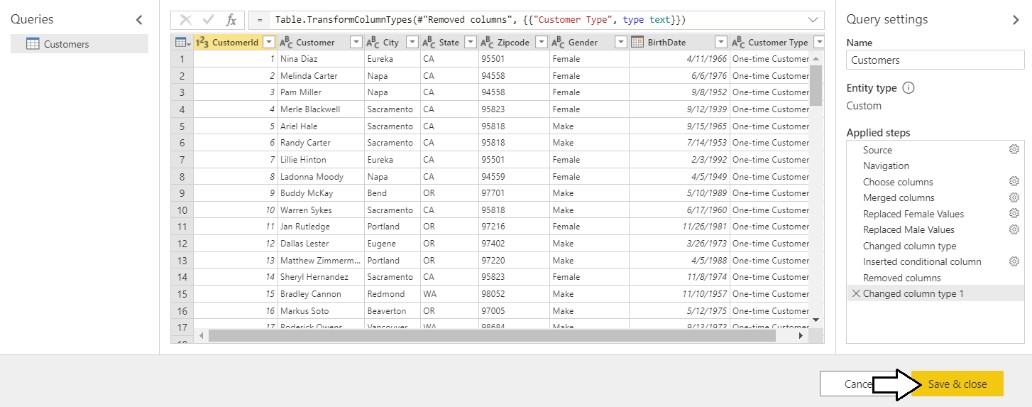
1. Set the column type for the **Customer Type** column to **Text**.
   1. You might notice that column type menu for the **Customer Type** column is not set to a specific type.



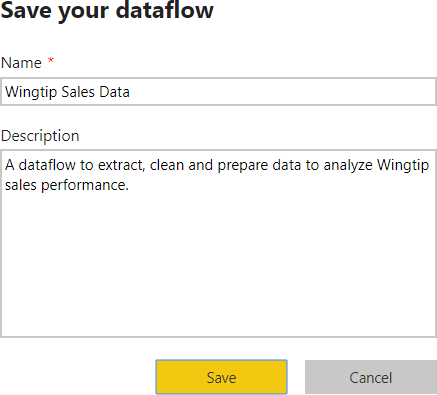
* 1. Drop down the Type menu for the **Customer Type** column and set its value to **Text**.



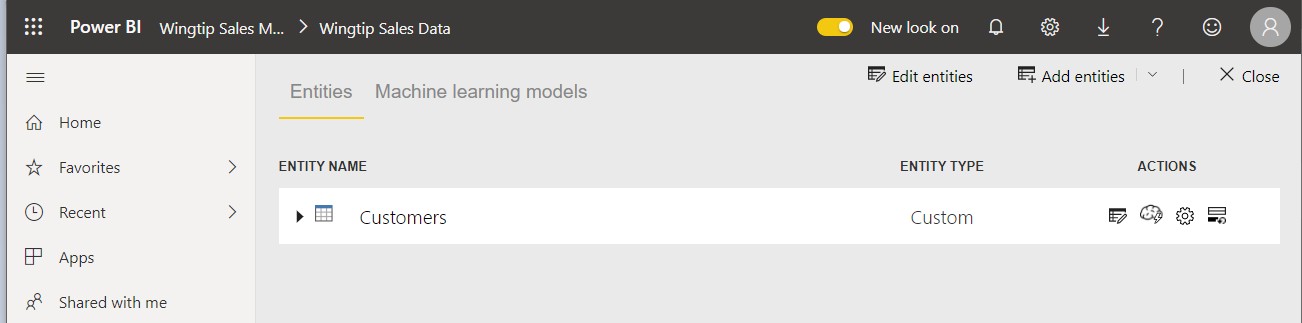
1. Save the dataflow with a name of **Demo Dataflow**.
   1. Click the **Save & close** button at the bottom right of the **Queries** window.



* 1. When prompted by the **Save your dataflow** dialog, enter a name of **Wingtip Sales Data** and then click **Save**.

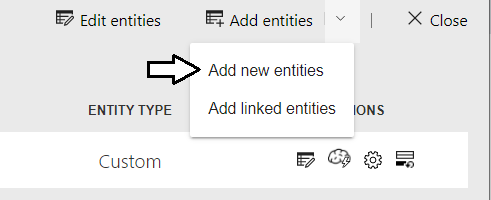


* 1. You should now see the summary page for the **Wingtip Sales Data** dataflow.

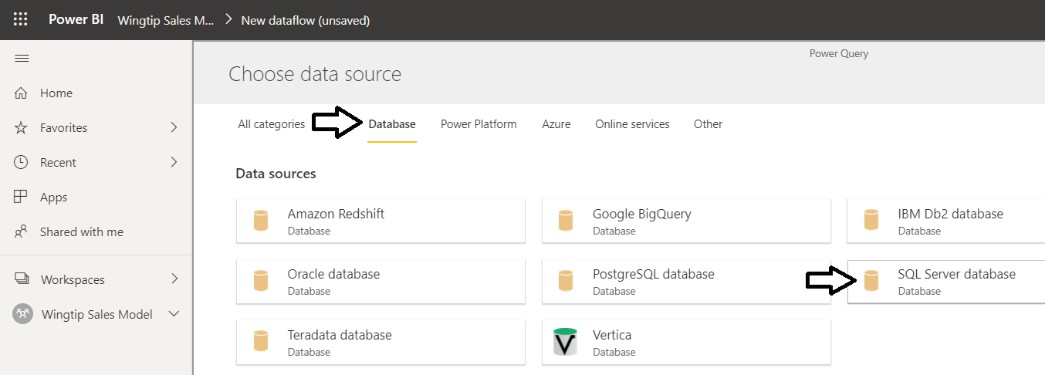


# Exercise 2: Extend the Dataflow by Adding Entities for Products and Sales

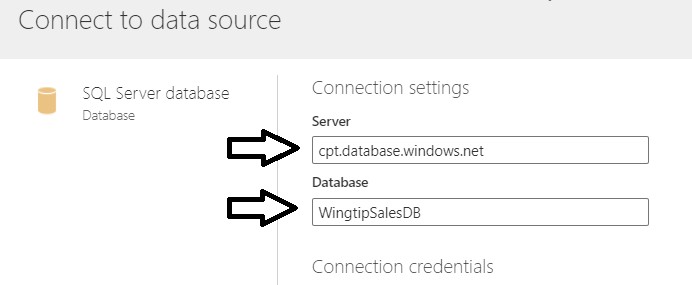
1. Aggiungere delle nuove entità
   1. Locate the **Add entities** menu button in the top right corner of the **Wingtip Sales Data** dataflow summary page.
   2. Drop down the **Add entities** menu button and select the **Add new entities** command.



* 1. On the **Choose data source** page, select the **Database** tab and then select **SQL Server database**.

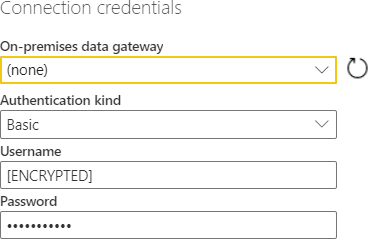


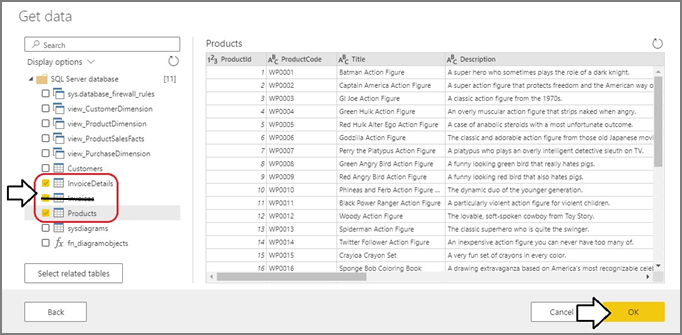
* 1. Enter a **Server** value of **afterhourdemo-ondemand.sql.azuresynapse.net**
  2. Enter a **Database** value of **Demo**.



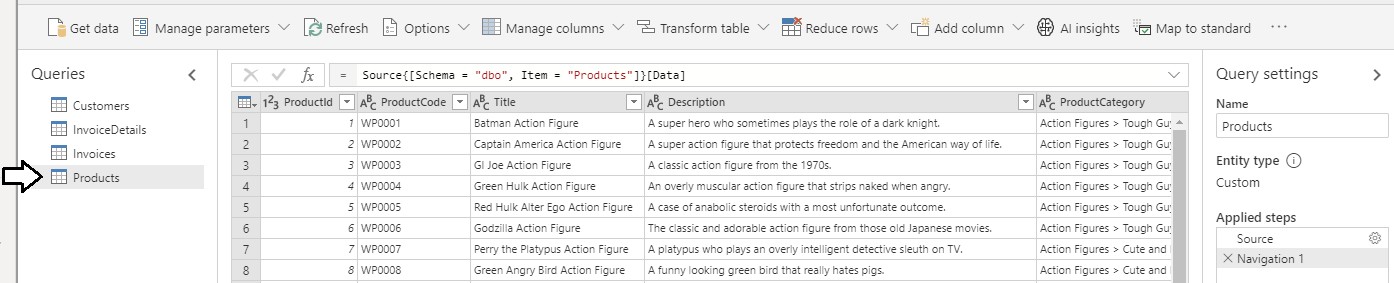
You should ***NOT*** be required to enter database credentials. That’s because you already entered the credentials for this database in exercise 1 and the credentials are now stored in the Microsoft cloud and will be used transparently.

* 1. Verify that values for **Username:** sqladminuserand **Password:** Pa$$w0rdwere automatically populated when you entered **Server** and **Database**.

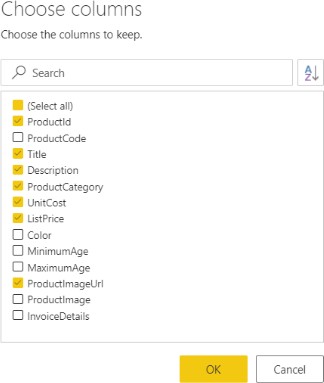


* 1. Click the **Next** button to continue.
  2. When prompted to **Choose data**, select the three tables named **InvoiceDetails** and **Products**.
  3. 
  4. Click the **OK** button to add three new entity queries to the dataflow.
  5. You should now see three new queries on the **Edit queries** page named **InvoiceDetails**, **Invoices** and **Products**.
  6. Select the **Products** query from the **Queries** list on the left so you can begin to modify its query logic with Power Query.

---------------------------------------------------Creao PRODUCTS DA INVOICES DETAILS -----------------------------------------------

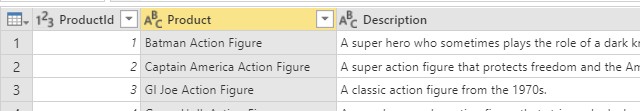


1. Remove the columns that are not required in the **Products** query results.
   1. Click the **Choose Columns** button in the ribbon to display the **Choose Columns** dialog.
   2. In the **Choose Columns** dialog, begin by clicking on the **(Select all)** checkbox at the top to unselect all columns.
   3. Select the checkboxes for **ProductId**, **Title**, **Description**, **ProductCategory**, **UnitCost**, **ListPrice** and **ProductImageUrl** as shown in the following screenshot.

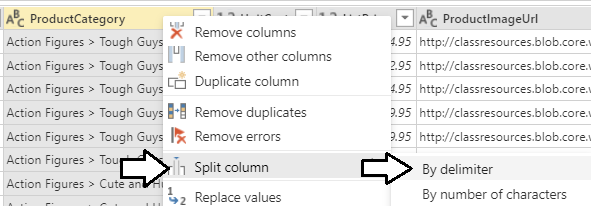


* 1. Click the **OK** button to close the **Choose Columns** dialog.

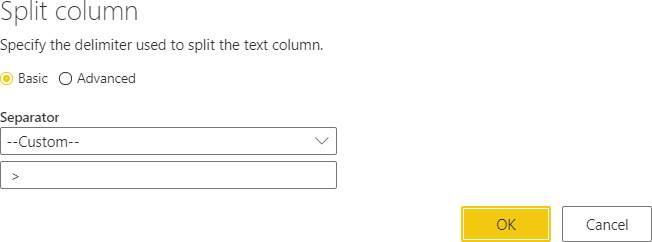
1. Rename the **Title** column to **Product**.
   1. Right-click on the **Title** column and click **Rename**.
   2. Update the column name to **Product**.



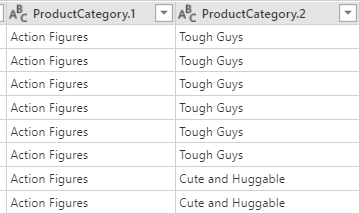
1. Split the **ProductCategory** column up into two separate columns named **Category** and **Subcategory**.
   1. Right-click the **ProductCategory** column and then click the **Split column** > **By delimiter** command.



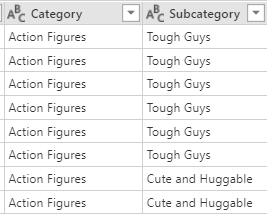
* 1. In the **Split column** dialog, drop down the **Separator** combo box and select **--Custom--**.
  2. In the textbox enter a three-character text value which includes a **space follow** by the **>** character followed by **another space**.
  3. When the **Split column** dialog matches the following screenshot, click the **OK** button.



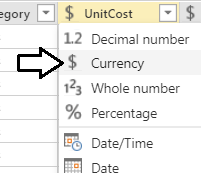
* 1. You should be able to confirm that Power BI Desktop has split the **ProductCategory** column into two separate columns named **ProductCategory.1** and **ProductCategory.2**.



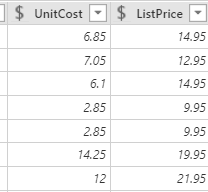
* 1. Rename the **ProductCategory.1** column to **Category** and rename **ProductCategory.2** to **Subcategory**.



1. Modify the column type of the **UnitCost** column and the **ListPrice** column to the **Currency** type.
   1. Use the dropdown column type menu to set the type of the **UnitCost** to **Currency**.



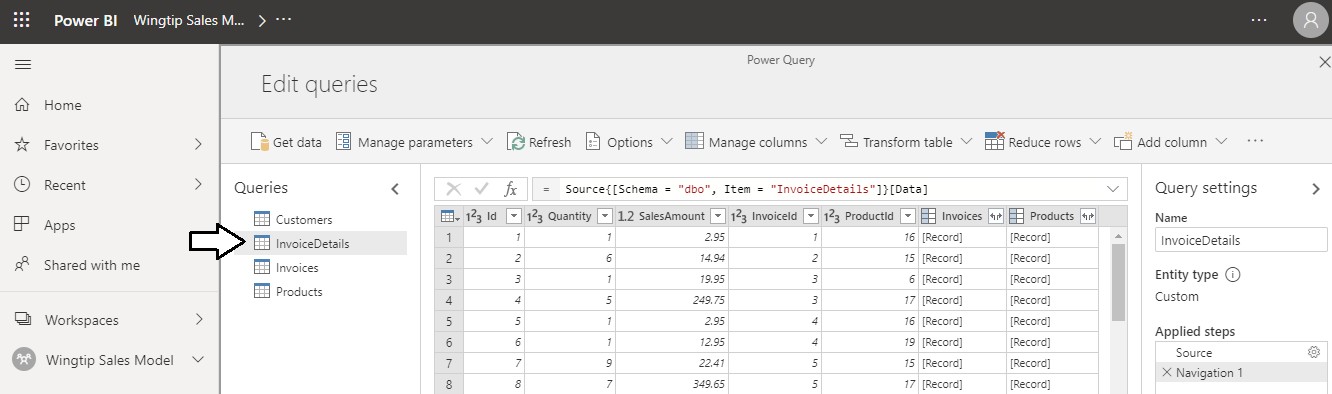
* 1. Use the dropdown column type menu to set the type of the **ListPrice** to **Currency**.



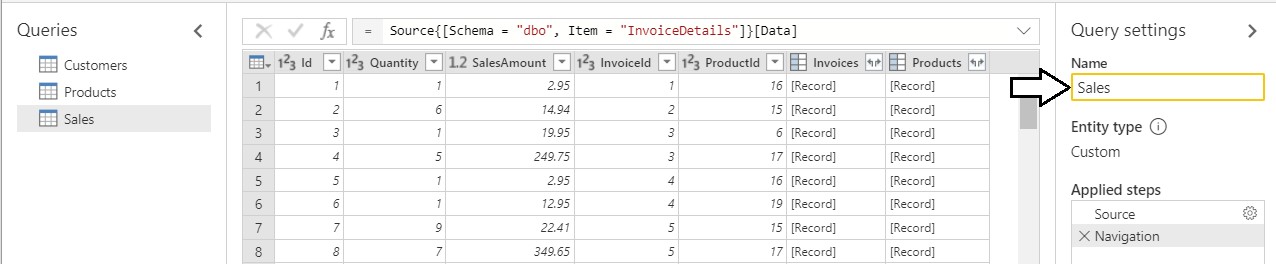
* 1. You have now completed your work on the **Products** query.

---------------------------------------------------Creao SALES DA INVOICES DETAILS -----------------------------------------------

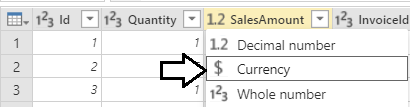
1. Rename the **InvoiceDetails** query to **Sales**.
   1. Select the **InvoiceDetails** query from the **Queries** list on the left.



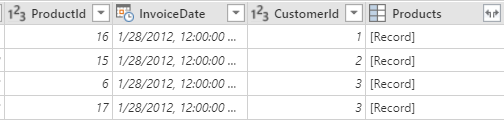
* 1. Update the name of the **InvoiceDetails** query to **Sales** by replacing the text in the **Name** textbox in the **Query Settings** pane.



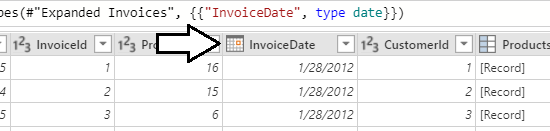
1. Modify the columns of the **Sales** query.
   1. Modify the column type of the **SalesAmount** column to the **Currency** type.



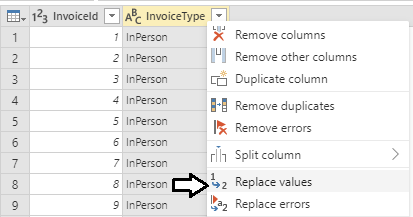
1. Expand the **Invoices** column to add the **InvoiceDate** column and the **CustomerId** column to the **Sales** query.
   1. Click the **Expand** button inside the column header of the **Invoices** column to display the **Columns to Expand** dialog.
   2. You should see that the **InvoiceDate** column and the **CustomerId** column have now been added to the **Sales** query results.



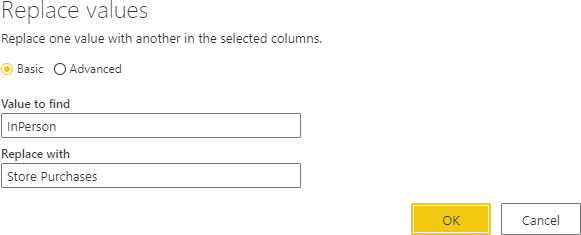
* 1. Change the column type of the **InvoiceDate** to the **Date** type.



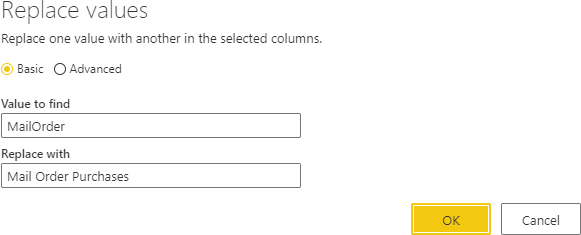
1. Modify the query so that the **InvoiceType** column returns values that are more human-readable.
   1. Right-click the header for the **InvoiceType** column and select **Replace values**.



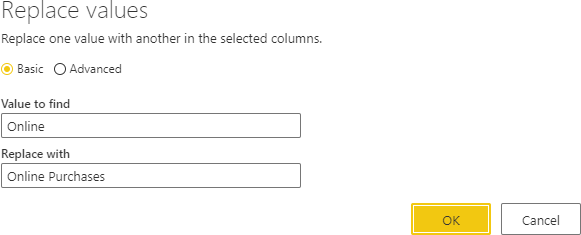
* 1. In the **Replace Values** dialog, enter a value of **InPerson** in the **Value to find** textbox.
  2. Enter a value of **Store Purchases** in the **Replace with** textbox and click **OK** to add your changes to the underlying query.



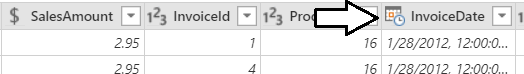
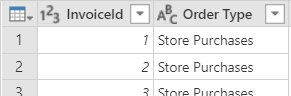
* 1. Add a second **Replace values** step to replace **MailOrder** with a value of **Mail Order Purchases**.



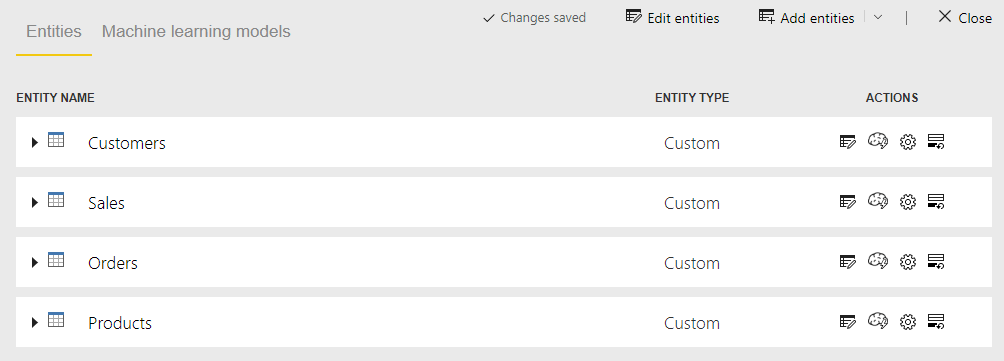
* 1. Add a third **Replace values** step to replace **Online** with a value of **Online Purchases**.



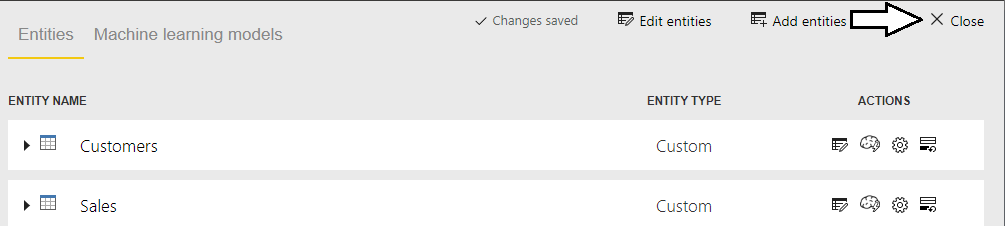
* 1. Change the name of the **InvoiceType** column to **Order Type**.



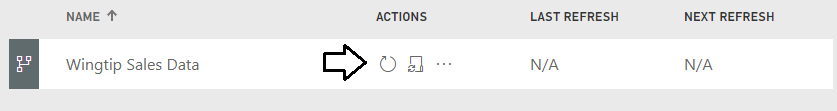
1. Save your work on workflow.
   1. Click the **Save & close** button at the bottom right of the page to save your work and return to the dataflow summary page.
   2. The **Entities** list should now display four entities named **Customers**. **Sales**, and **Products**.



1. Refresh the dataflow to populate it with data.
   1. Click the **Close** button in the upper right corner of the dataflow summary page.



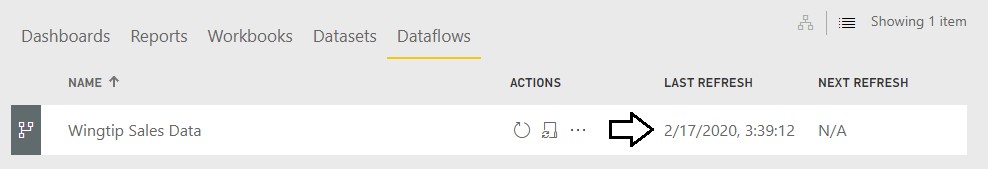
* 1. You should now see the **Wingtip Sales Data** dataflow in the app workspace summary page.
  2. Click the Refresh button to begin a refresh operation on the **Wingtip Sales Data** dataflow.



* 1. Wait for the refresh operation to complete. It might take one or two minutes to complete.



* 1. Once the refresh operation completes, you should see the **LAST REFRESH** time has been updated.

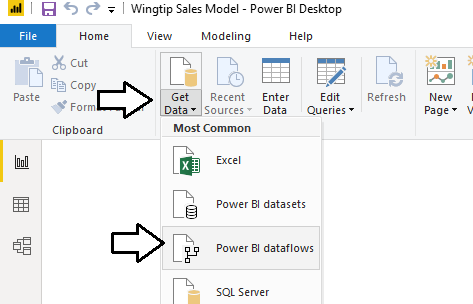


# Exercise 3: Importing Dataflow Entity Data with Power BI Desktop

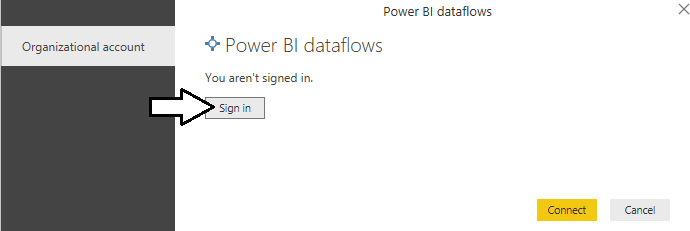
1. Launch Power BI Desktop to start a new project.
2. Save the new project as **Dataflow.pbix** using the following path.

**C:\Student\Projects\Dataflow.pbix**

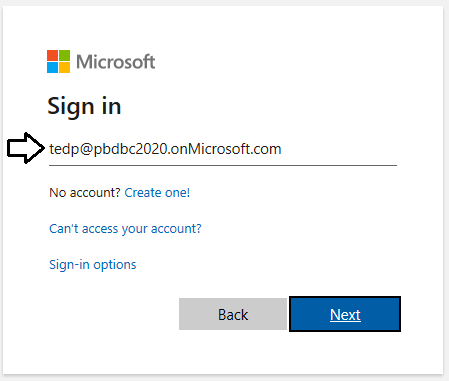
1. Import all four entities from the **Wingtip sales Data** dataflow.
   1. Drop down the **Get Data** menu button on the ribbon and click **Power BI dataflows**.



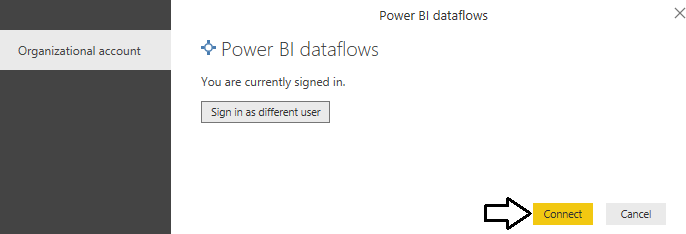
* 1. When prompted by the **Power BI dataflows** connect dialog, click **Sign in**.



* 1. In the **Sign in** dialog, enter your account name and then click **Next**.

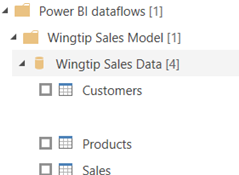


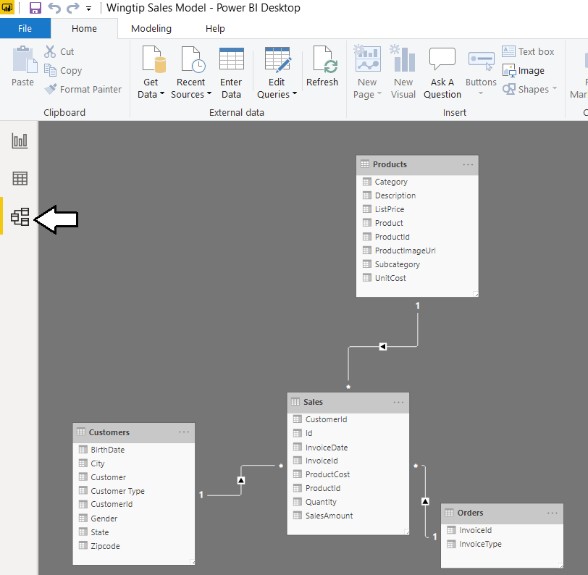
* 1. Once you have signed in, click the **Connect** button.



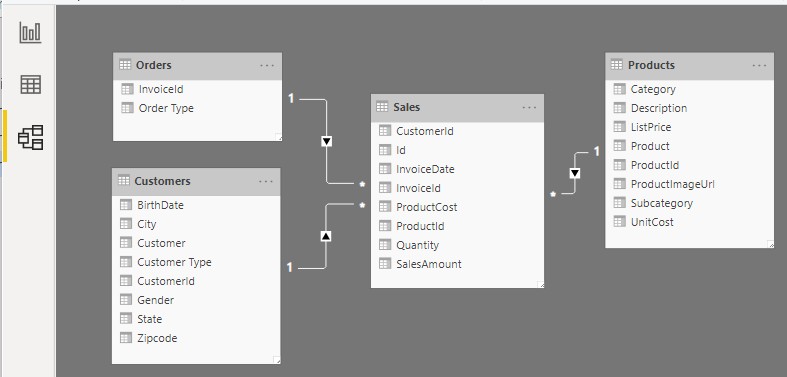
* 1. In the **Navigator** dialog, expand the **Wingtip Sales Model** workspace



* 1. Expand the **Wingtip Sales Data** workflow so you can see the entities inside this dataflow.
  2. Once the queries execute, click the **Model View** button to see the four tables imported from the dataflow.

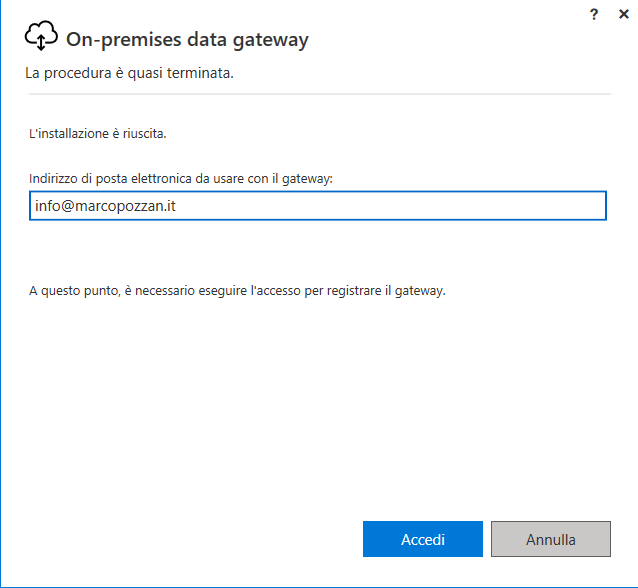


* 1. Using the mouse, rearrange the four tables in Model View to match the following screenshot.

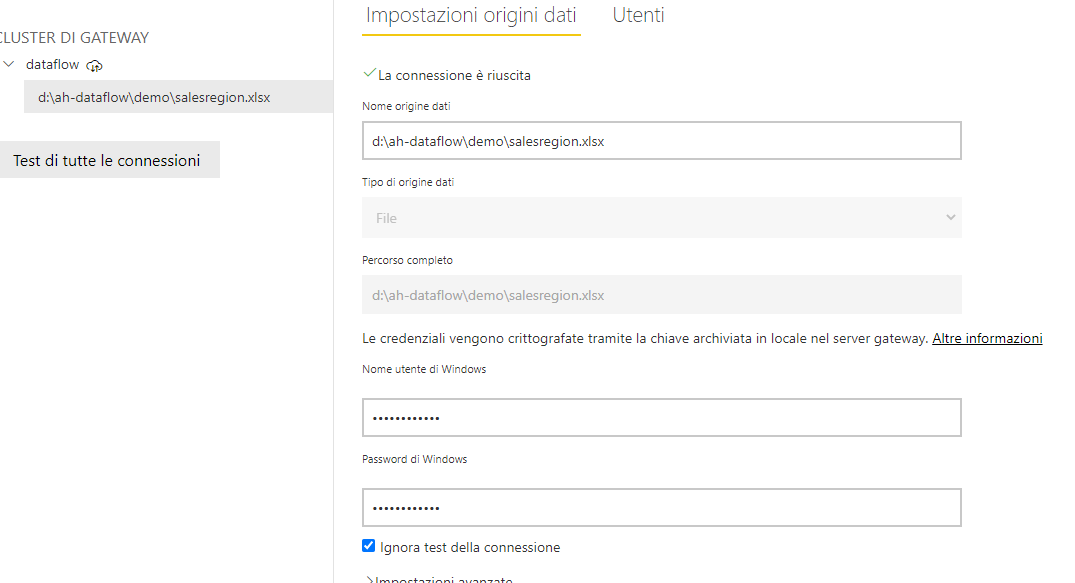


* 1. Save your work to **dataflow.pbix**.

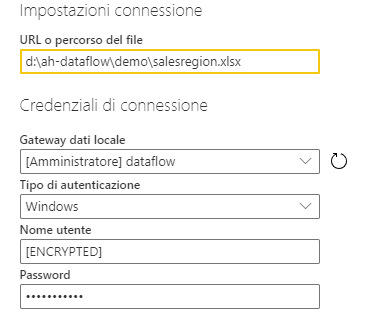
1. Aggiungiamo il **salesregion nel data flow che è preso da un file excel in locale ad una macchina**
   1. Mostrare che non ha il gateway e quindi non posso prendere il file.
   2. Installare il gateway



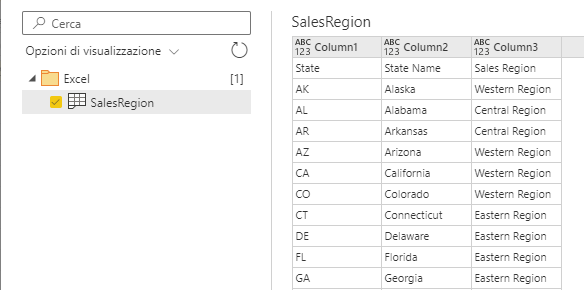
* 1. Andiamo a configurare il percorso nel gateway username: **admin** e password: la solita **mia**



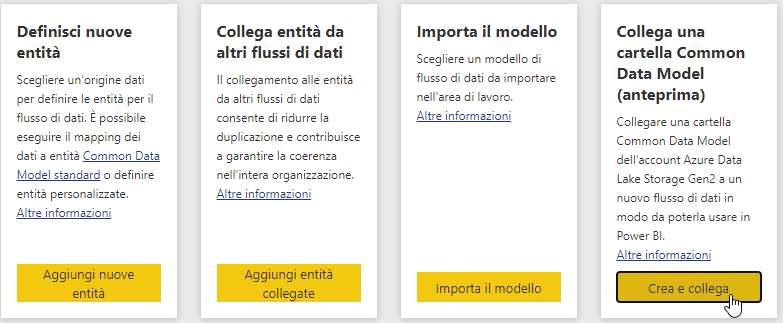
* 1. Poi selezioniamo il file dal dataflow e in automatico riconscerà il percorso di rete



* 1. Poi incomincia il trasferiento dati



Demo 3



Spiegare il modello model.json https://adlsafterhour.dfs.core.windows.net/synapse/model.json

{"name":"Demo Dataflow",

 "description":"",

 "version":"1.0",

 "culture":"it-IT",

 "entities":[{"$type":"LocalEntity",

              "name":"Calendario",

              "description":"",

              "attributes":[{"name":"Date","dataType":"date"},

                            {"name":"Year","dataType":"int64"},

                            {"name":"Quarter","dataType":"string"},

                            {"name":"Month","dataType":"string"},

                            {"name":"MonthSort","dataType":"date"},

                            {"name":"Month in Year","dataType":"string"},

                            {"name":"MonthInYearSort","dataType":"int64"},

                            {"name":"Day of Week","dataType":"string"},

                            {"name":"DayOfWeekSort","dataType":"int64"}

                            ],

                "partitions":[{"name":"1","location":" https://adlsafterhour.dfs.core.windows.net/synapse/Calendario.csv"}]

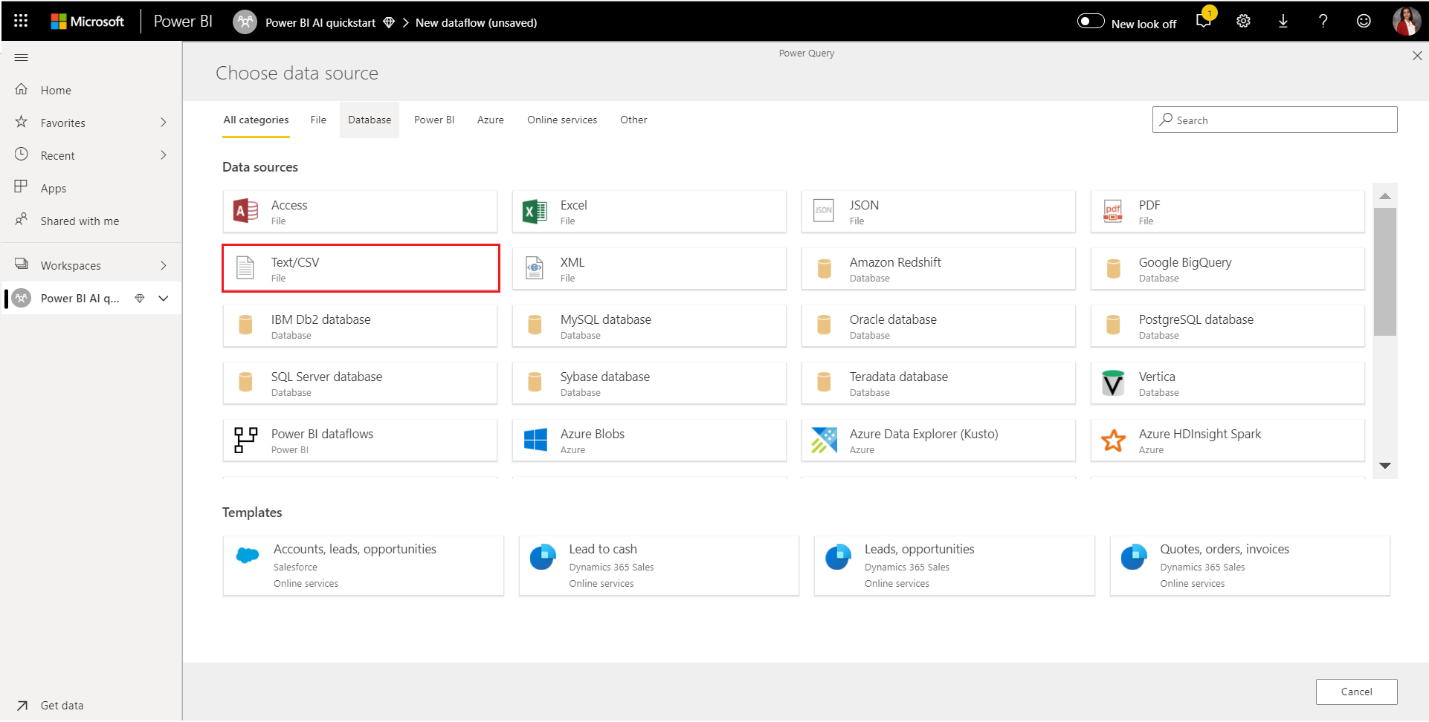
                }]}

Demo 4

Select **Add new entities**. This launches a **Power Query** editor in the browser.

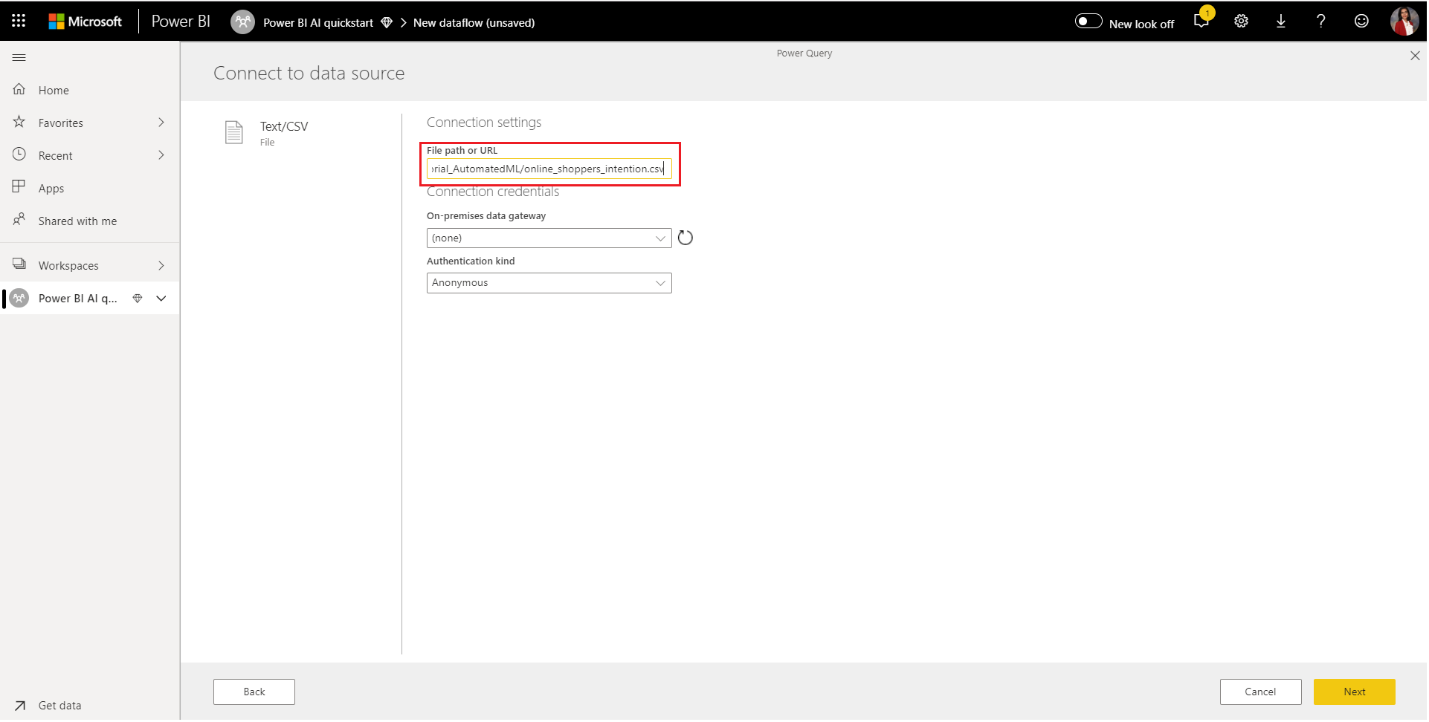


Select **Text/CSV File** as a data source, shown in the following image.

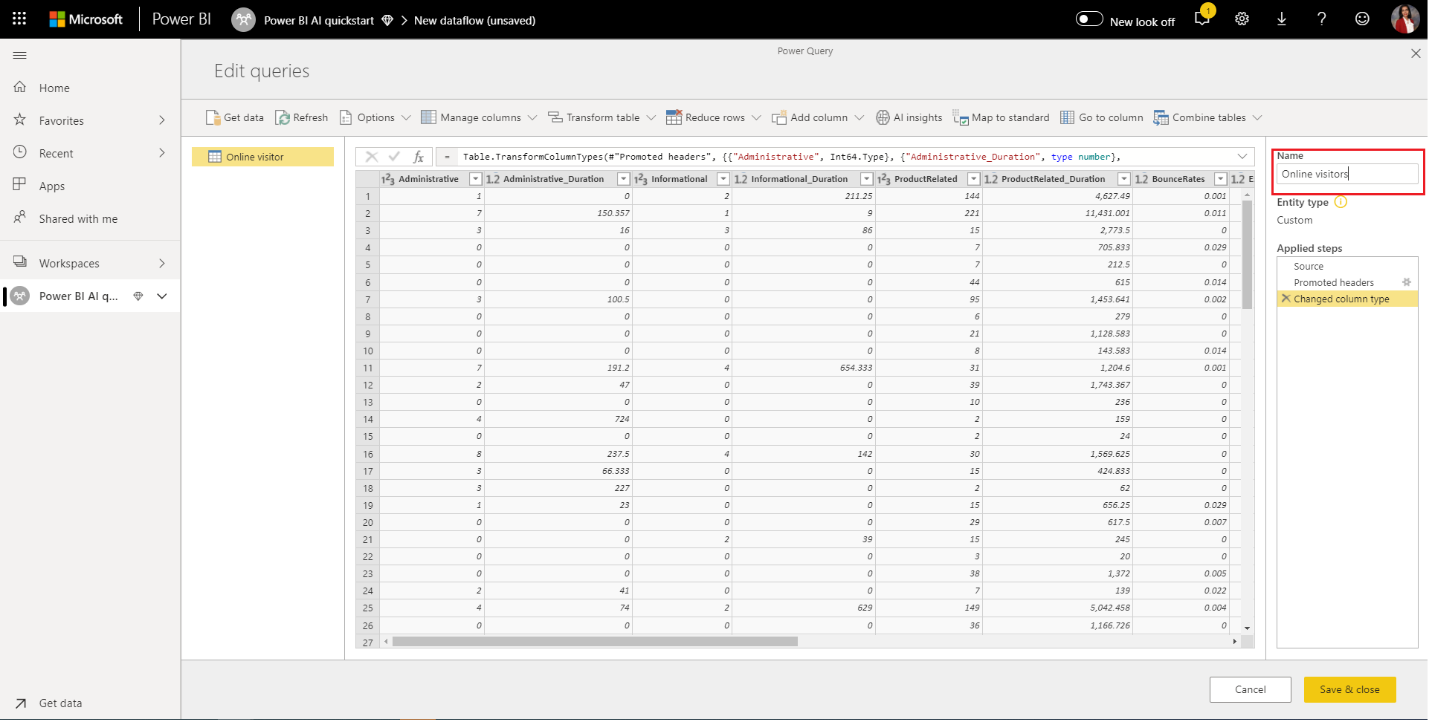


In the **Connect to a data source** page that appears next, paste the following link to the online\_shoppers\_intention.csv into the **File path or URL** box, and then select **Next**.

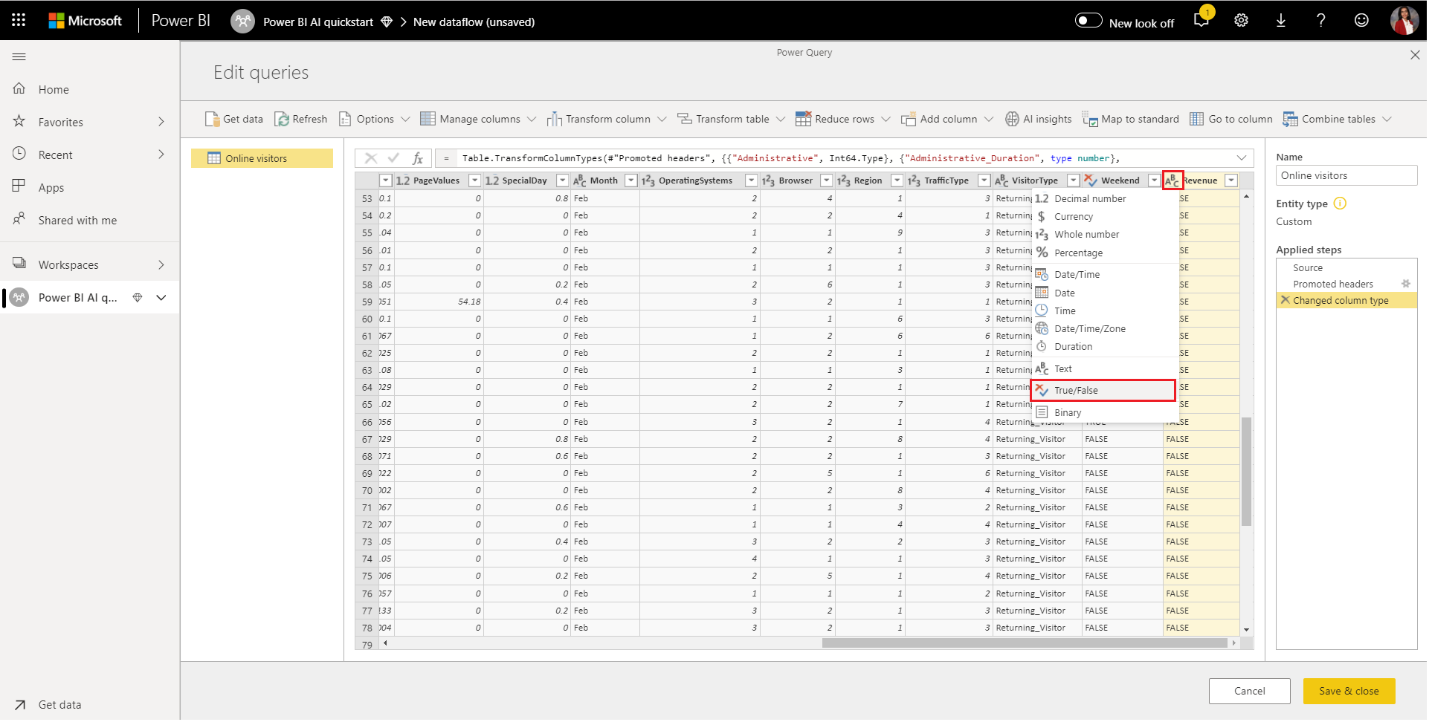
https://raw.githubusercontent.com/santoshc1/PowerBI-AI-samples/master/Tutorial\_AutomatedML/online\_shoppers\_intention.csv



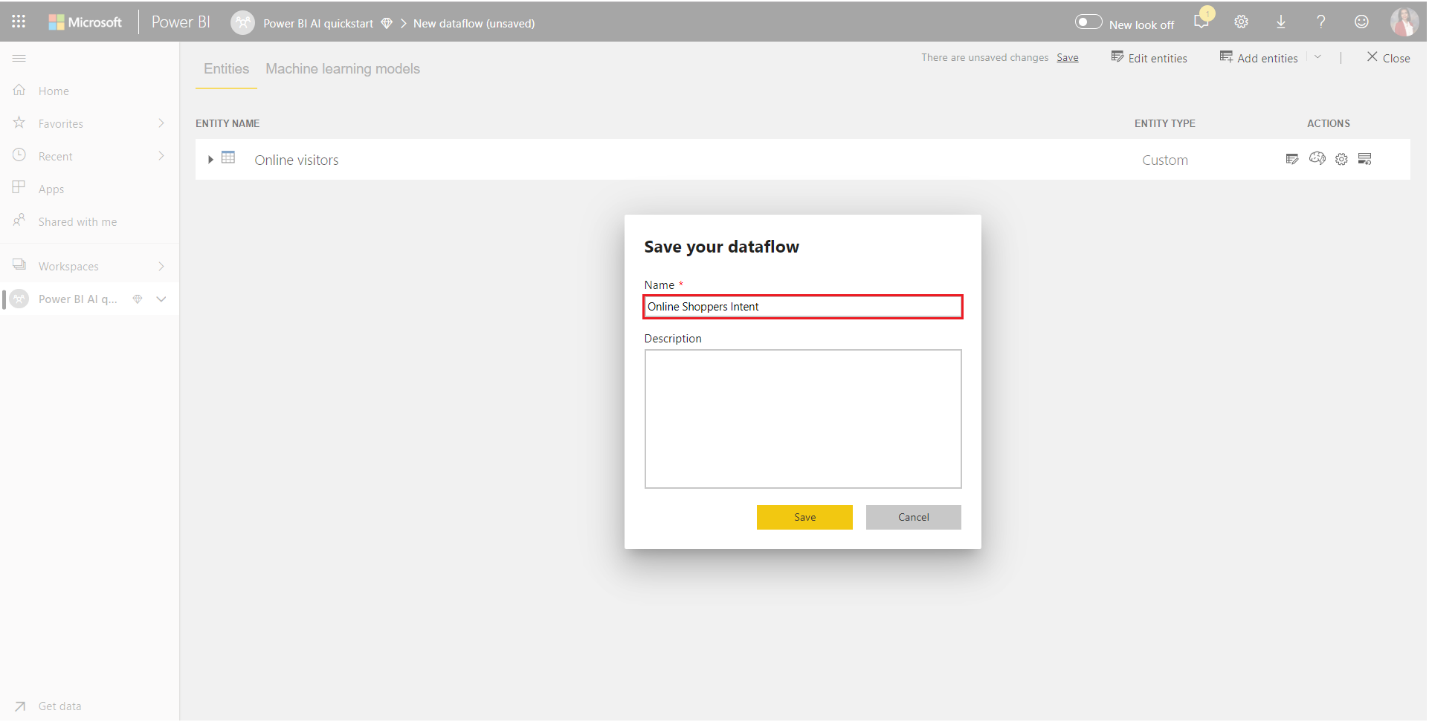
The Power Query Editor shows a preview of the data from the CSV file. You can rename the query to a friendlier name by changing the value in the Name box found in the right pane. For example, you could change the Query name to **Online Visitors**.



Power Query automatically infers the type of columns. You can change the column type by clicking on the attribute type icon at the top of the column header. In this example, we change the type of the Revenue column to True/False.

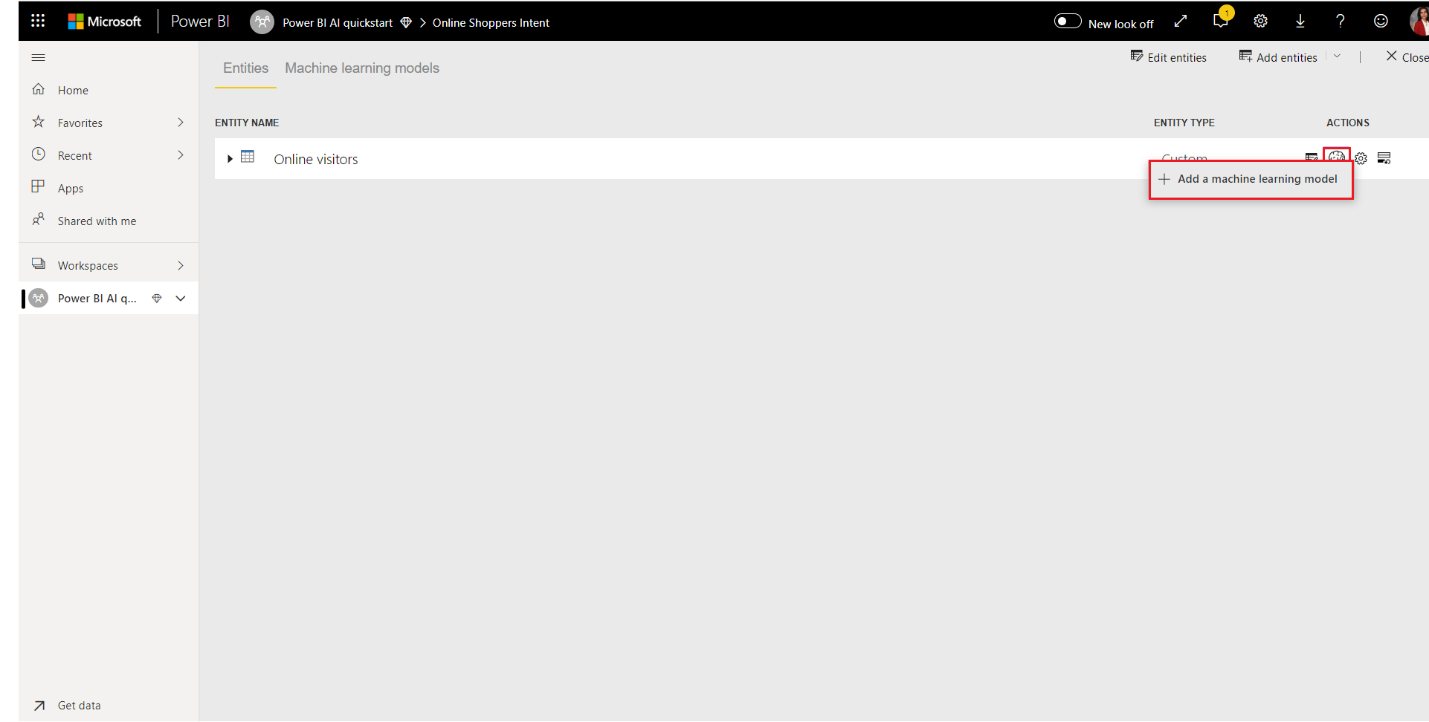


Select the **Save & close** button to close Power Query Editor. Provide a name for the dataflow, and then select **Save** on the dialog, as shown in the following image. **Online Shoppers Intent**



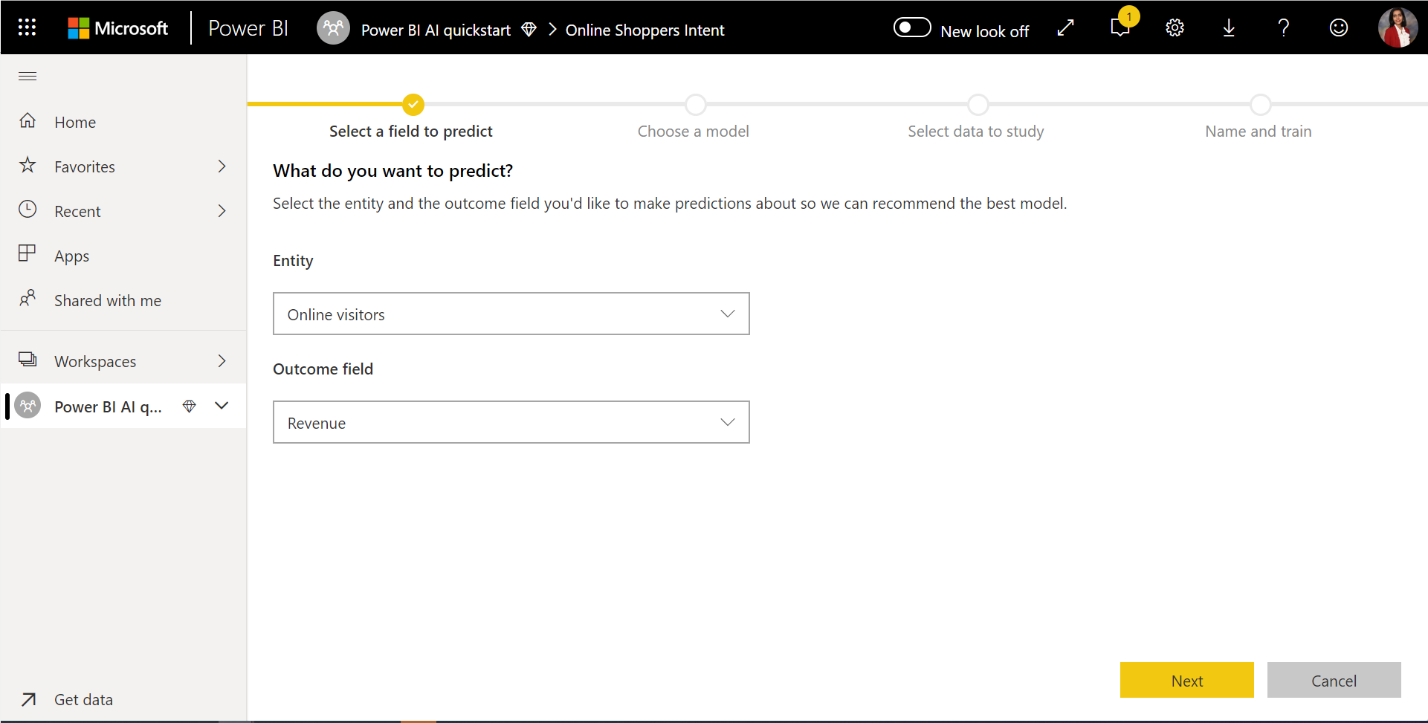
## **Create and train a machine learning model**

To add a machine learning model, Select the **Apply ML model** button in the **Actions** list for the base entity that contains your training data and label information, and then select **Add a machine learning model**.

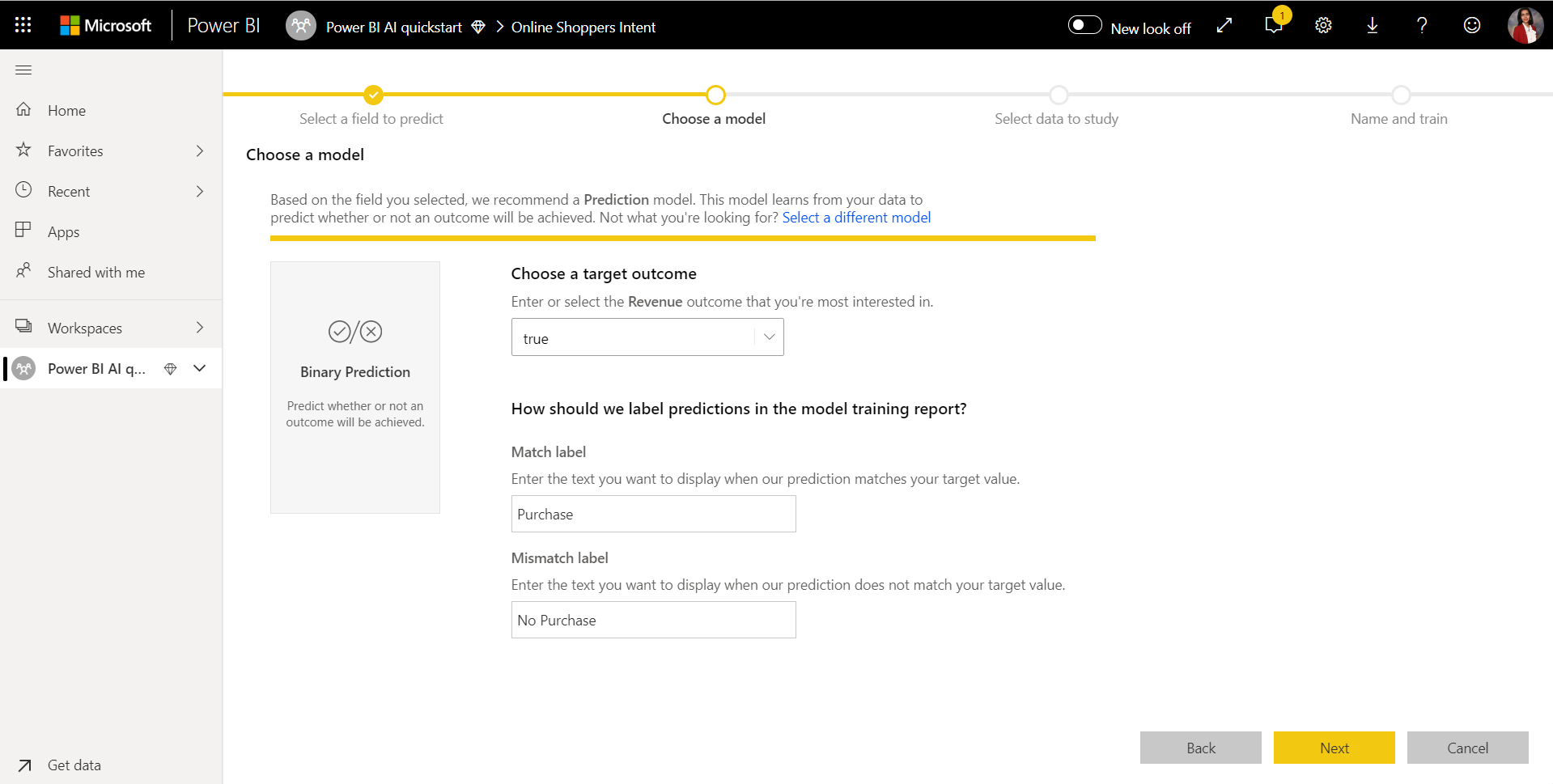


The first step for creating our **machine learning model is to identify the historical data including the outcome field that you want to predict.** The model will be created by learning from this data.

In the case of the dataset we're using, this is the **Revenue** field. Select **Revenue** (**Valore che vogliamo predire**) as the 'Outcome field' value and then select **Next**.



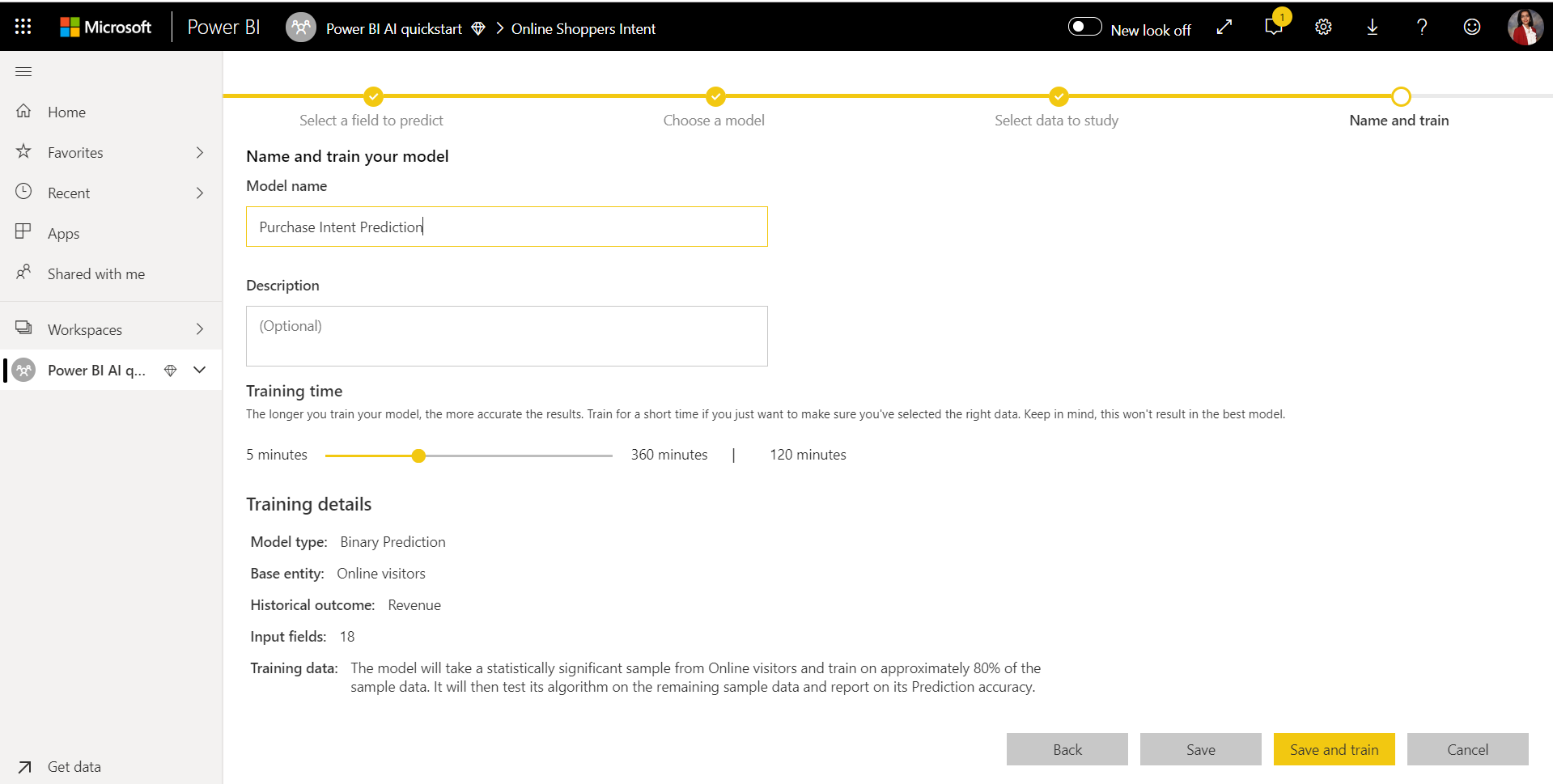
Next, we must select the type of machine learning model to create. Power BI analyzes the values in the outcome field that you've identified and suggests the types of machine learning models that can be created to predict that field.

In questo caso, poiché prevediamo un risultato binario che indica se un utente effettuerà un acquisto o meno, si consiglia la previsione binaria. Poiché siamo interessati a prevedere gli utenti che effettueranno un acquisto, seleziona Vero come risultato delle entrate che ti interessa di più. Inoltre, puoi fornire etichette amichevoli per i risultati da utilizzare nel rapporto generato automaticamente che riepiloga i risultati della validazione del modello. Quindi seleziona Avanti.

Next, Power BI does a preliminary scan of a sample of your data and suggests the inputs that may produce more accurate predictions. If Power BI doesn't recommend a field, an explanation would be provided next to it. You have the option to change the selections to include only the fields you want the model to study, or you can select all the fields by selecting the checkbox next to the entity name. Select **Next** to accept the inputs.

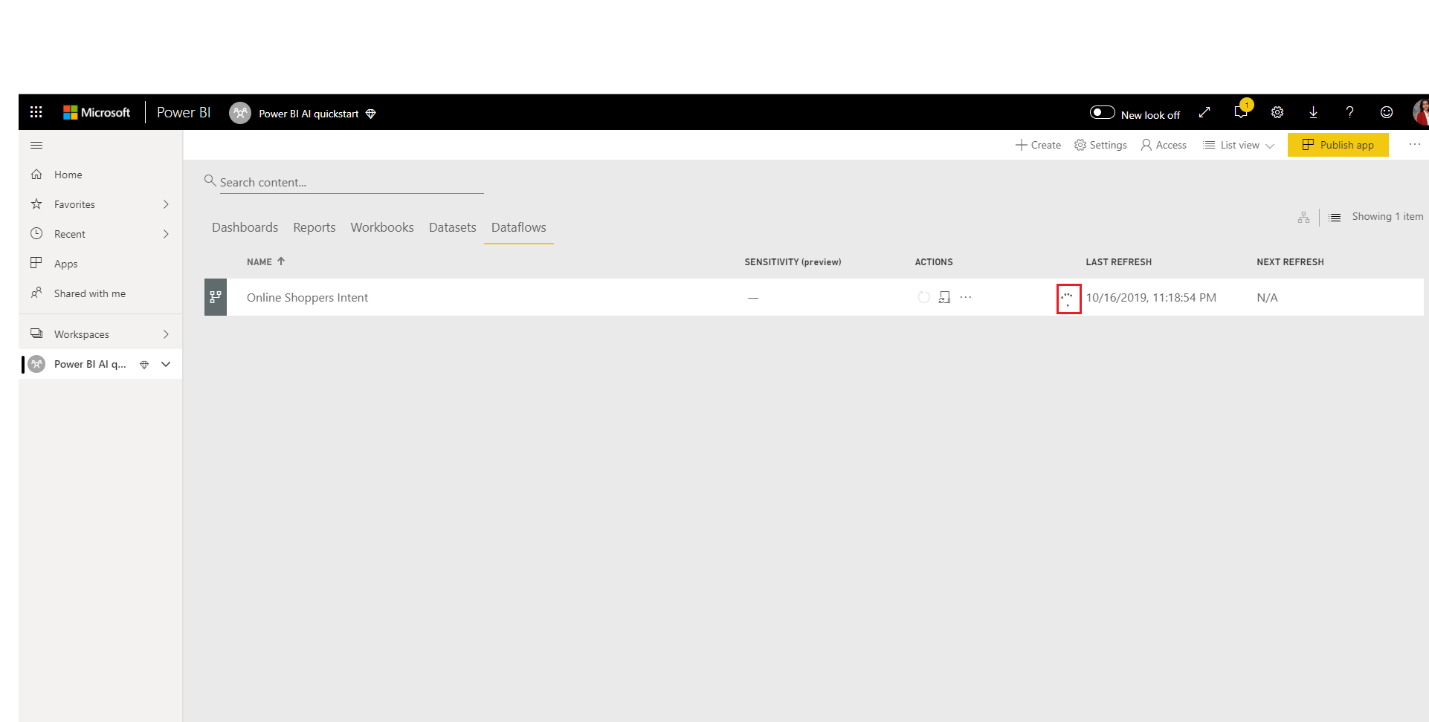


In the final step, we must provide a name for our model. Name the **model Purchase Intent Prediction.** You can choose to reduce the training time to see quick results or increase the amount of time spent in training to get the best model. Then select **Save and train** to start training the model.

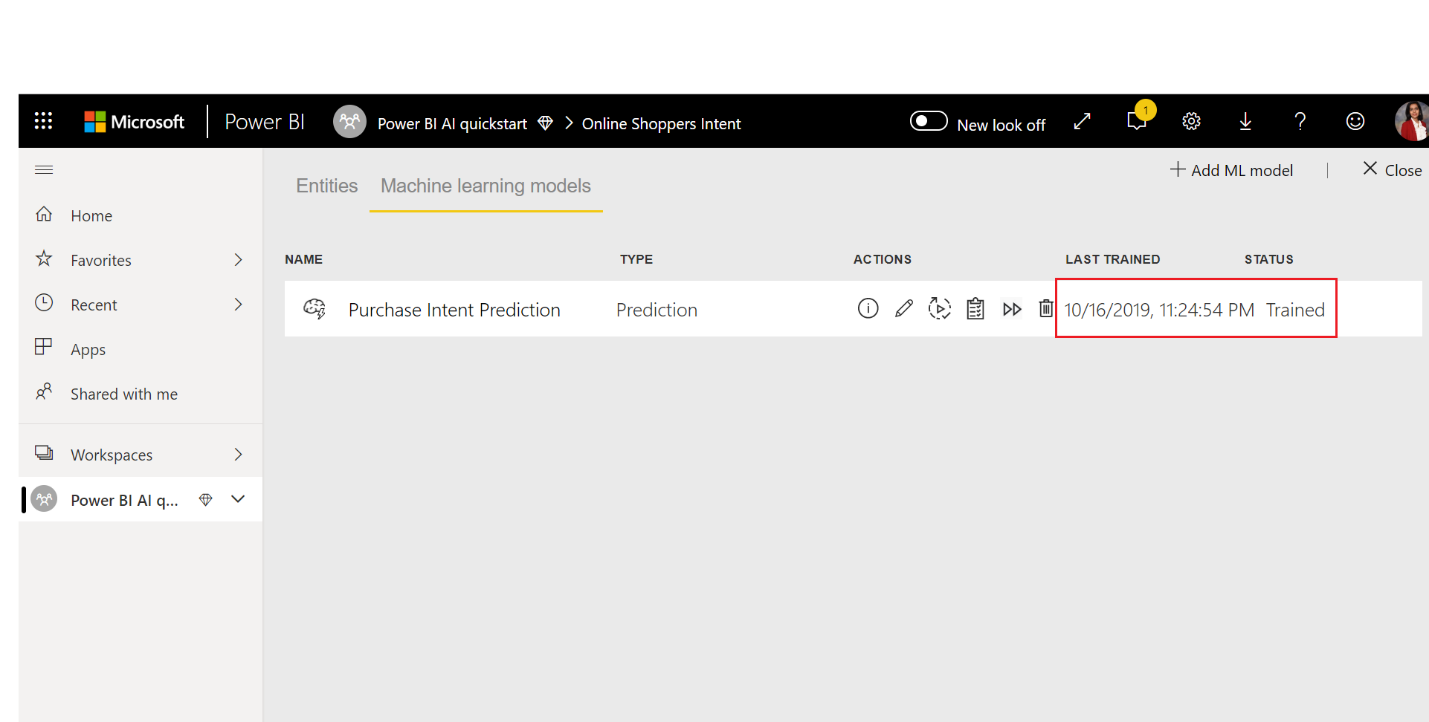


Il processo di formazione inizierà campionando e normalizzando i dati cronologici e suddividendo il set di dati in due nuove **entità Dati di formazione sulla previsione dell'intenzione di acquisto e** Dati di **test sulla previsione dell'intenzione di acquisto**.

You can confirm that the model is being trained and validated through the status of the dataflow. This appears as a data refresh in progress in the **Dataflows** tab of the workspace.



Once the model training is completed, the dataflow displays an updated refresh time. You can confirm that the model is trained, by navigating to the **Machine learning models** tab in the dataflow. The model you created should show status as **Trained** and the **Last Trained time** should now be updated.

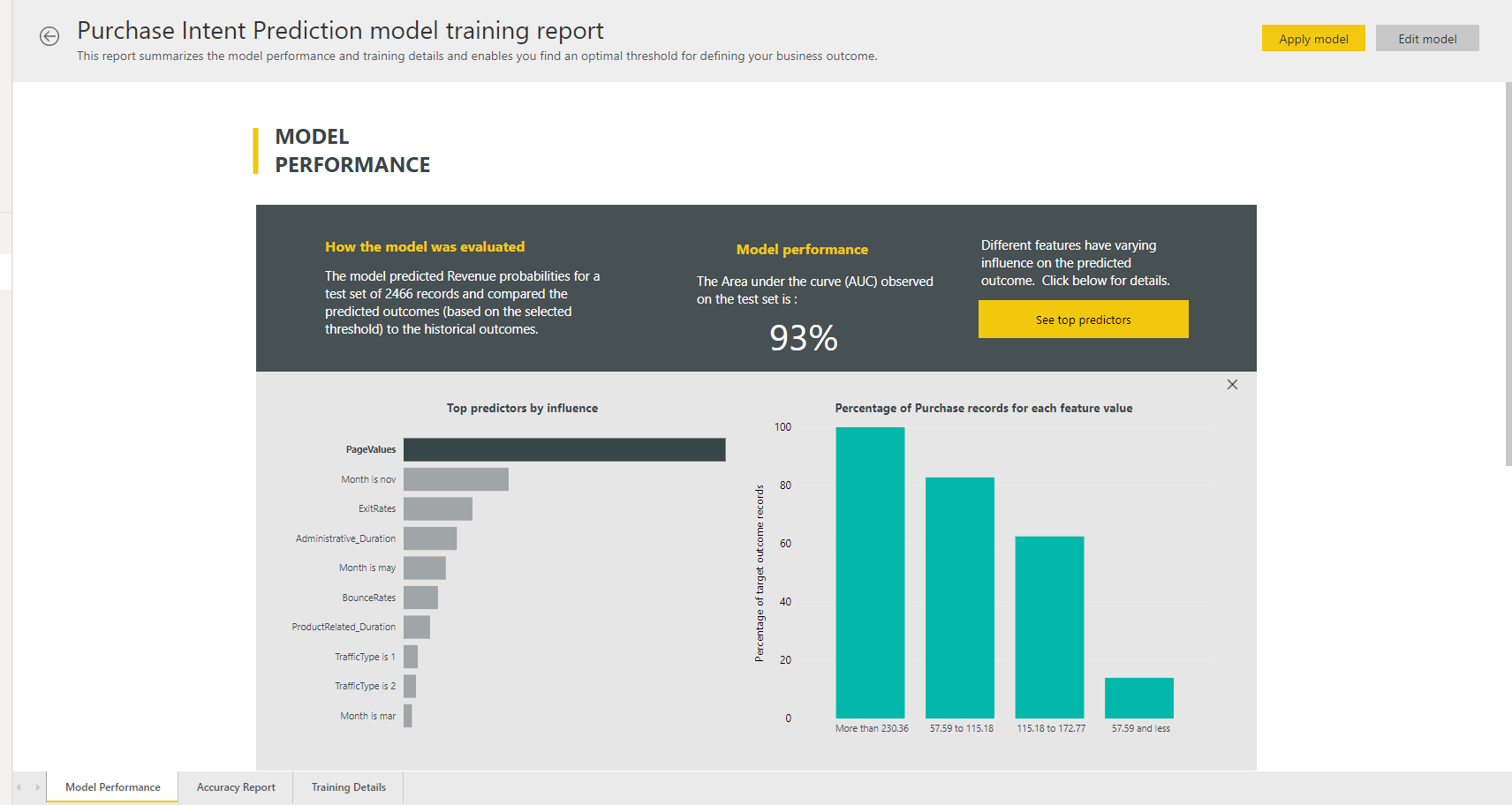


## **Review the model validation report**

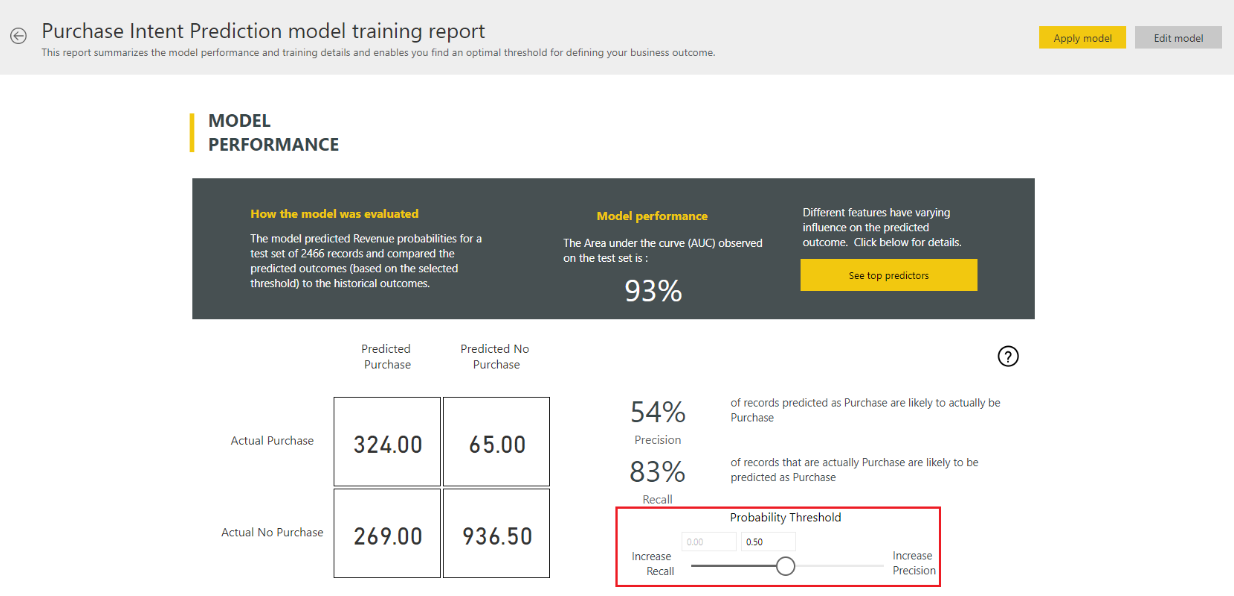
To review the model validation report, in the Machine learning models tab, select the View training report button in the Actions column for the model. This report describes how your machine learning model is likely to perform.

In the **Model Performance** page of the report, select See **top predictors** to view the top predictors for your model. You can select one of the predictors to see how the outcome distribution is associated with that predictor.





You can use the **Probability Threshold** slicer on the Model Performance page to examine its influence on the Precision and Recall for the model.

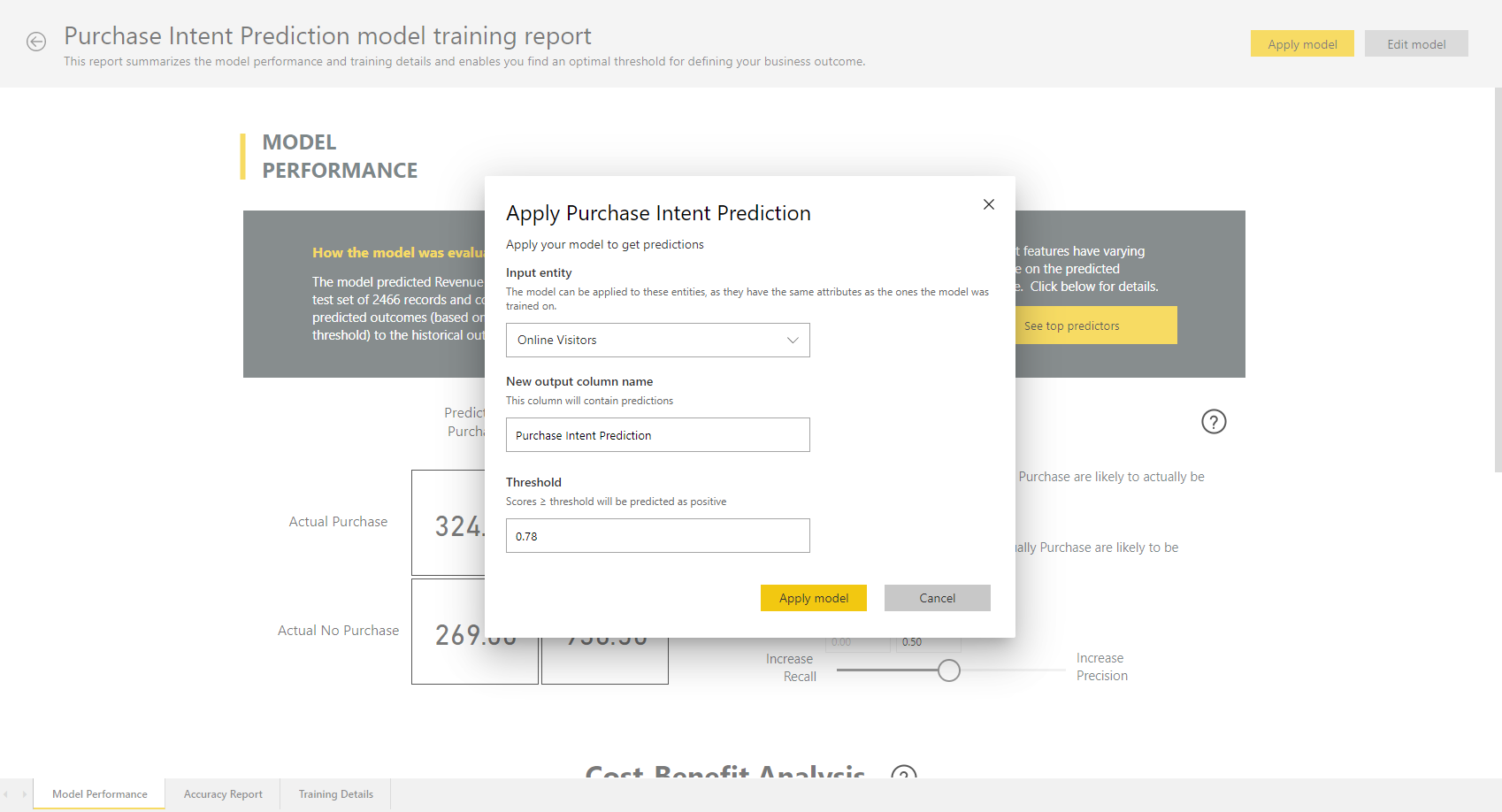


The other pages of the report describe the statistical performance metrics for the model.

The report also includes a Training Details page that describes the different iterations that were run, how features were extracted from the inputs, and the hyperparameters for the final model used.

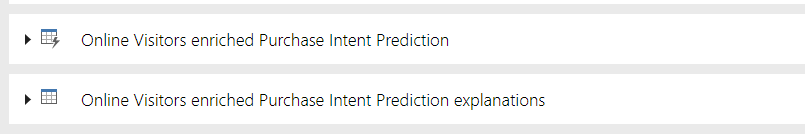
## Apply the model to a dataflow entity

Select the **Apply model** button at the top of the report to invoke this model. In the **Apply** dialog, you can specify the target entity that has the source data to which the model should be applied.



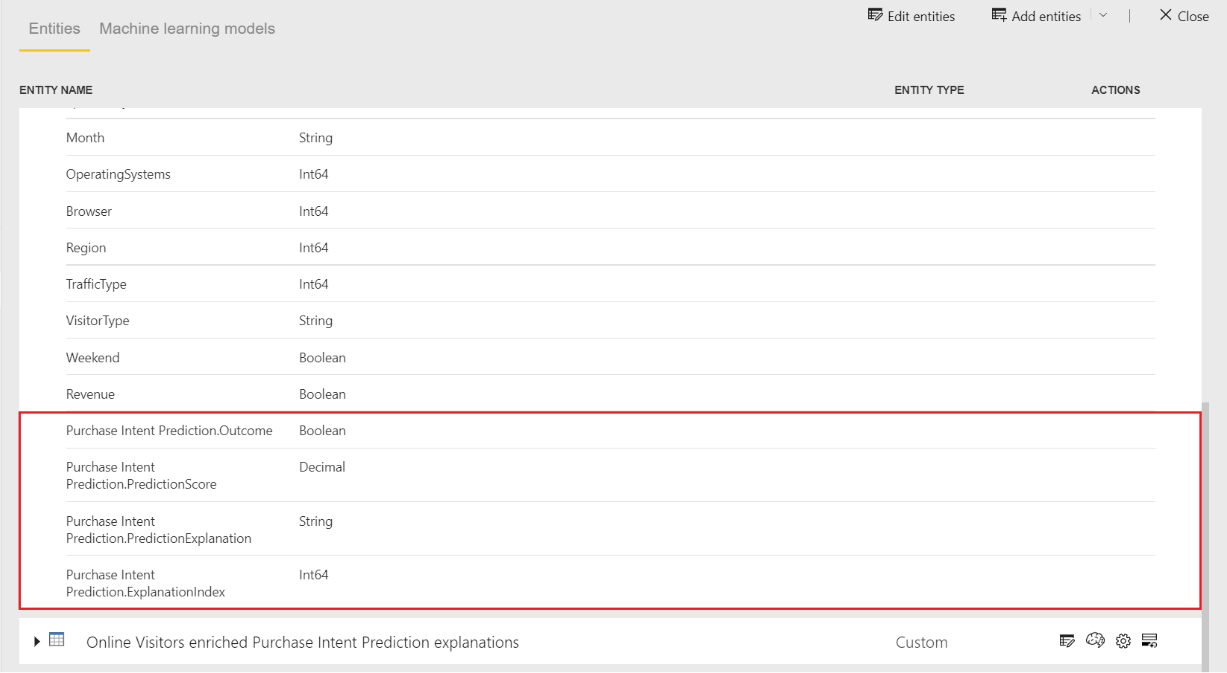
When prompted, you must **Refresh** the dataflow to preview the results of your model.

Applying the model will create two new entities, with the suffix **enriched <model\_name>** and **enriched <model\_name> explanations**.

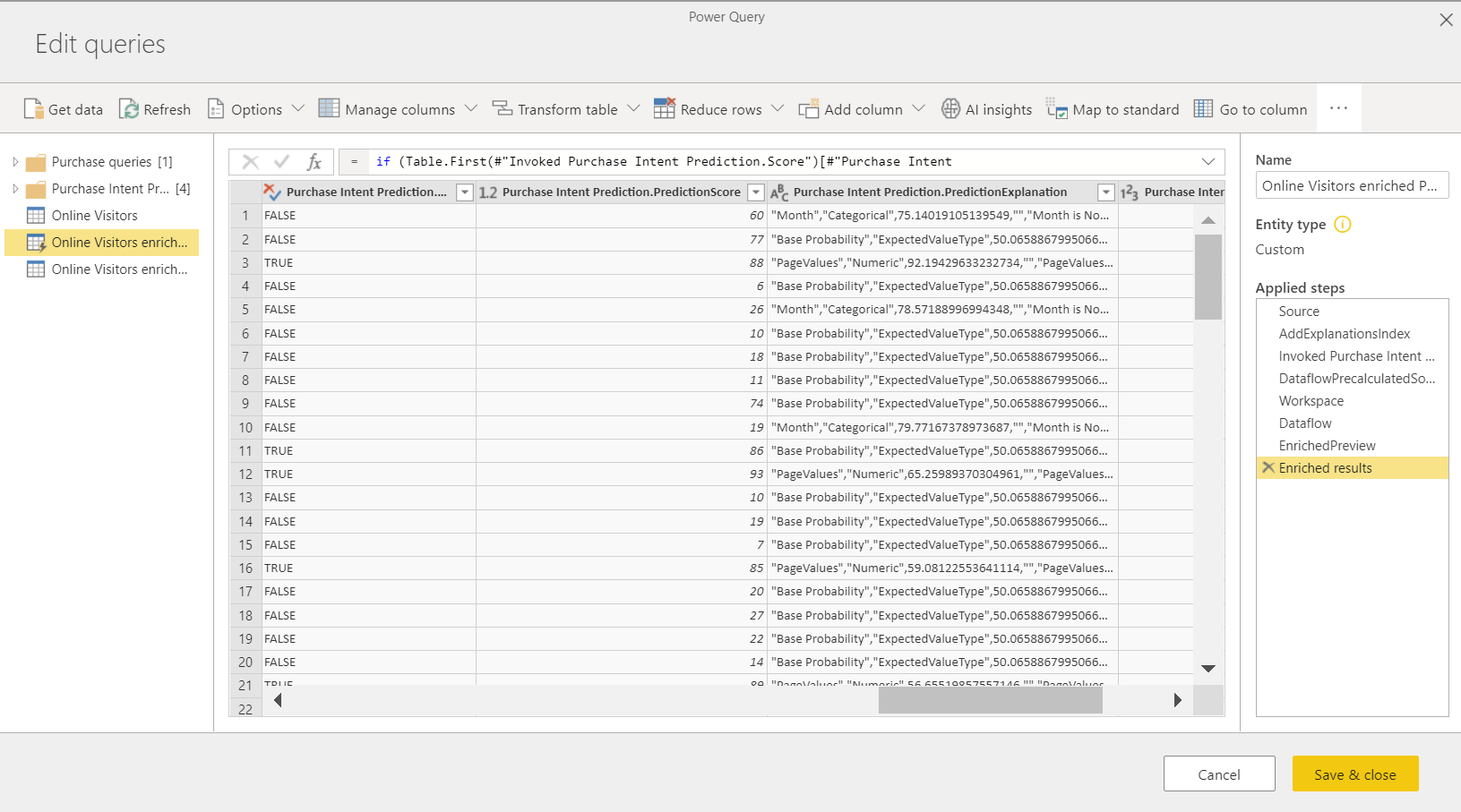


In our case, applying the model to the **Online Visitors** entity will create **Online Visitors enriched Purchase Intent Prediction** (che include l'output previsto dal modello ), and **Online Visitors enriched Purchase Intent Prediction explanations** (che contiene i principali influencer specifici del record per la previsione.)

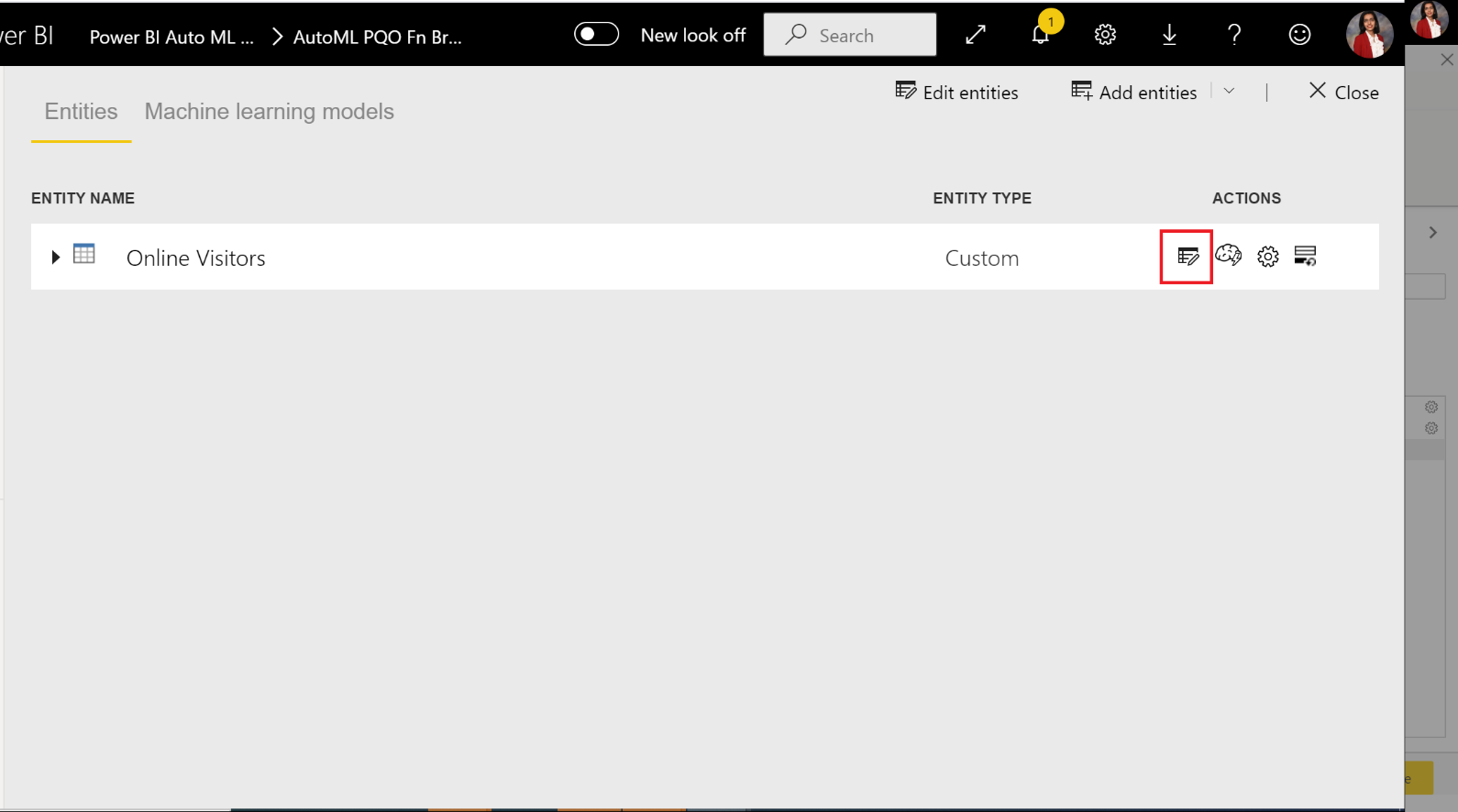
Applying a Binary Prediction model adds 4 columns with predicted outcome, probability score, the top record-specific influencers for the prediction, and explanation index each prefixed with the column name specified.



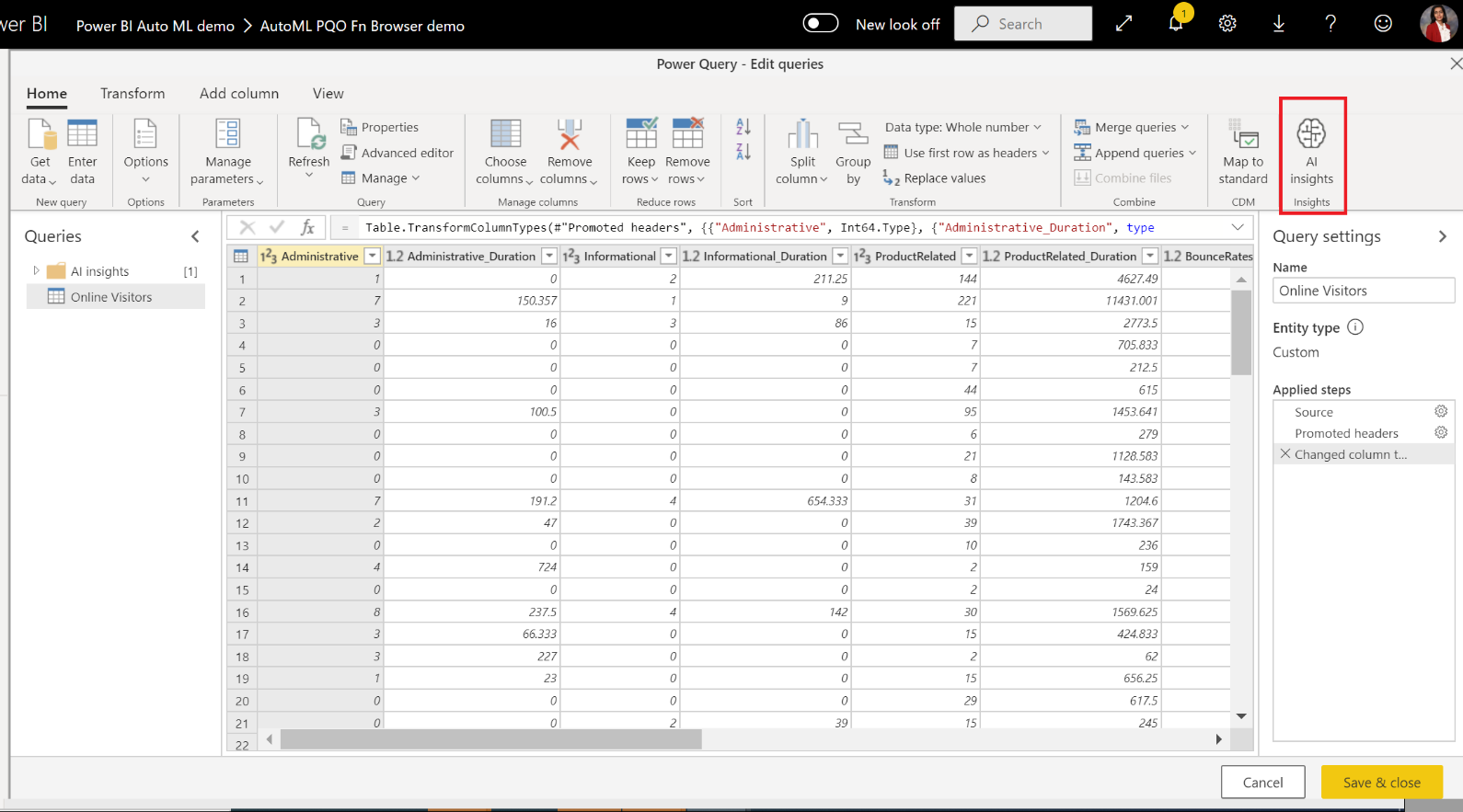
Once the dataflow refresh is completed, you can select the **Online Visitors enriched Purchase Intent Prediction** entity to view the results.



You can also invoke any AutoML model in the workspace, directly from the Power Query Editor in your dataflow. To access the AutoML models, select the Edit button for the entity that you want to enrich with insights from your AutoML model, as shown in the following image.

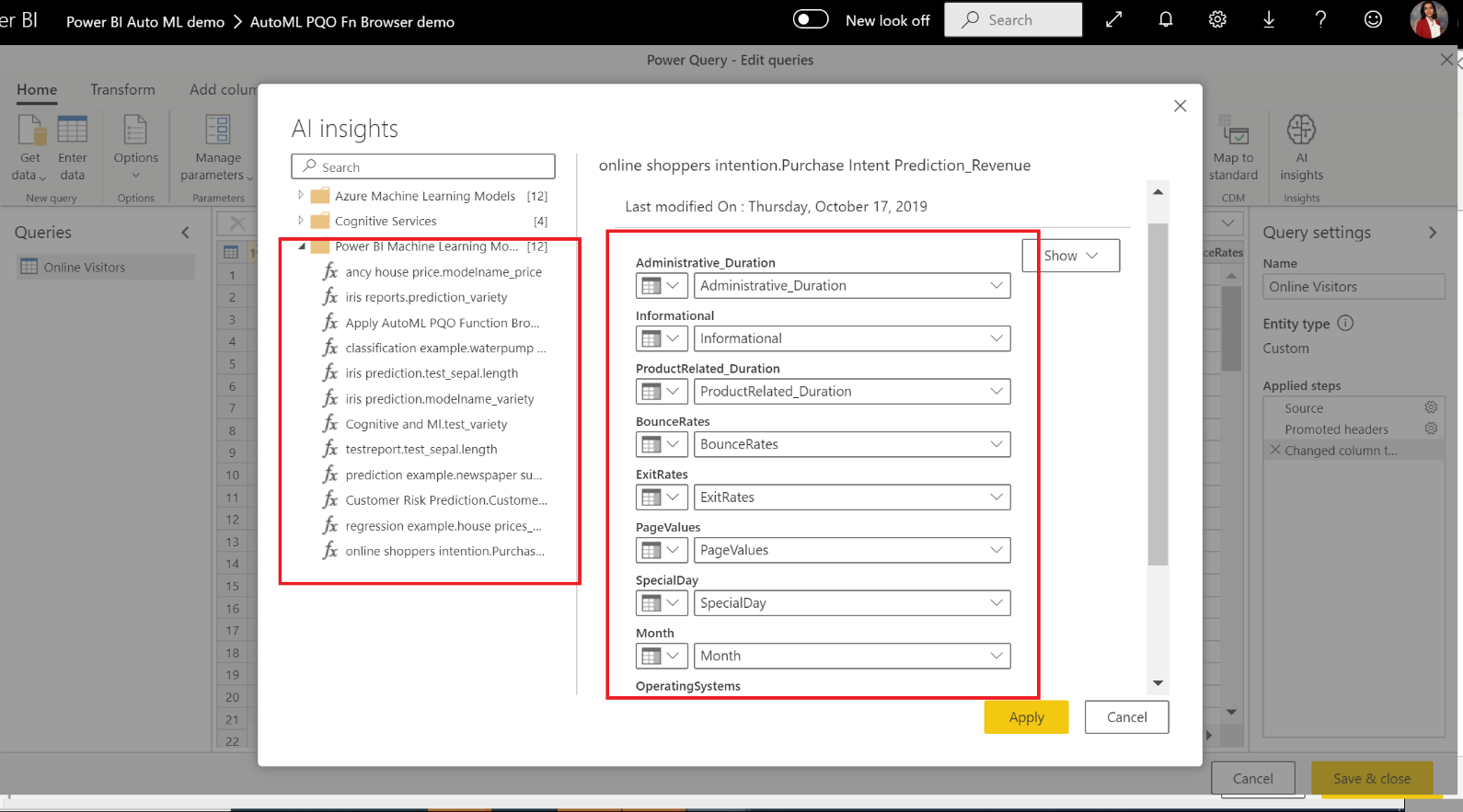


Selecting the Edit button opens the Power Query Editor for the entities in your dataflow. Select the AI Insights button in the ribbon.

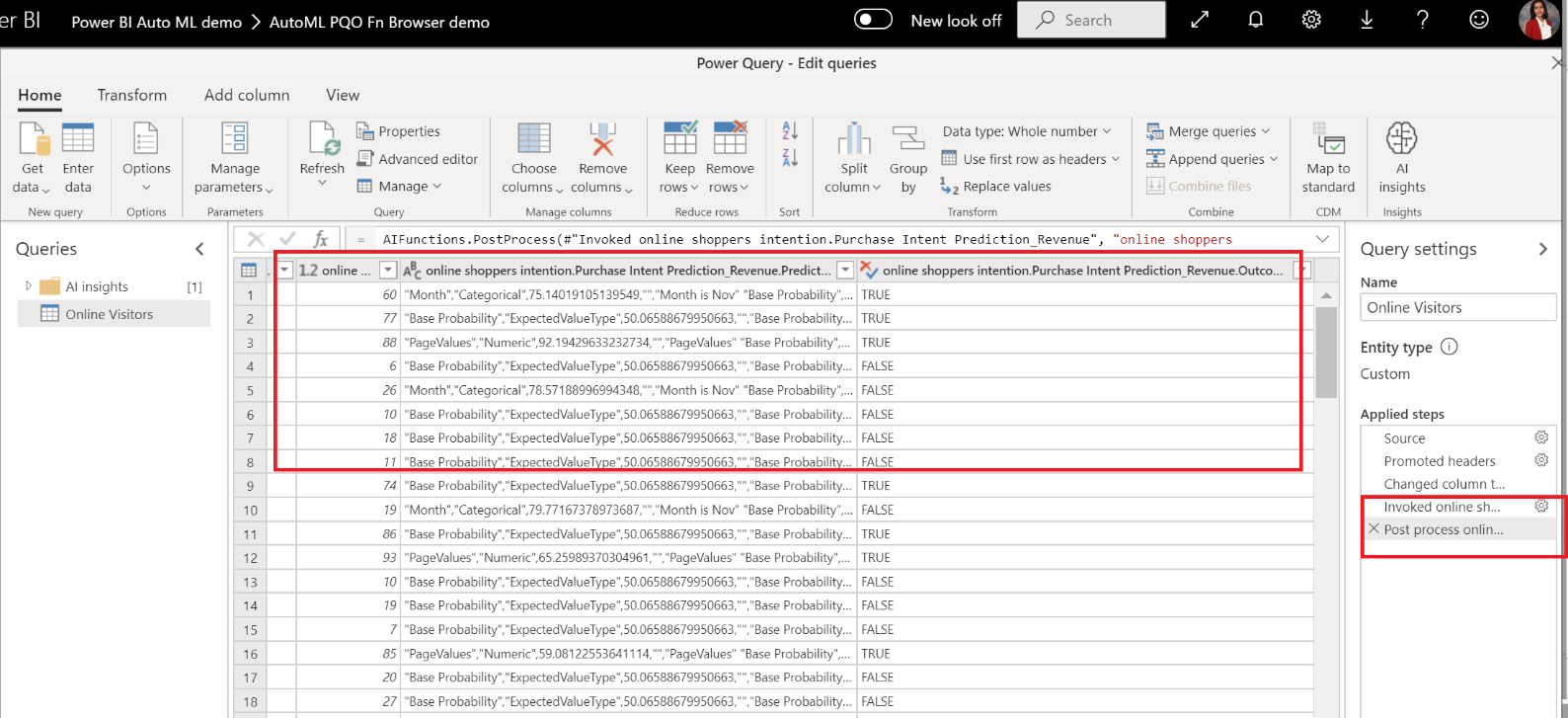


Select the Power BI Machine Learning Models folder from the nav pane menu. All the AutoML models to which you have access are listed here as Power Query functions. Also, the input parameters for the AutoML model are automatically mapped as parameters of the corresponding Power Query function. Note that automatic mapping of parameters happens only if the name and data type of the parameter is the same.

To invoke an AutoML model, you can specify any of the selected entity's columns as an input from the drop-down. You can also specify a constant value to be used as an input by toggling the column icon to the left of the input dialog.



Select Apply to view the preview of the AutoML model's output as a new columns in the entity table. You will also see the model invocation as an applied step for the query.



Once you save your dataflow, the model is automatically invoked when the dataflow is refreshed, for any new or updated rows in the entity table.

create VIEW dbo.InvoiceDetails\_temp

AS

SELECT \*

FROM

openrowset(

        BULK N'https://adlsafterhour.dfs.core.windows.net/synapse/PBI\_Course\_InvoiceDetails.csv',

        FORMAT = 'CSV',

        PARSER\_VERSION='2.0',

        HEADER\_ROW = TRUE ,

        firstrow = 1

    ) as x;

create VIEW dbo.Invoice

AS

SELECT \*

FROM

openrowset(

        BULK N'https://adlsafterhour.dfs.core.windows.net/synapse/PBI\_Course\_Invoices.csv',

        FORMAT = 'CSV',

        PARSER\_VERSION='2.0',

        HEADER\_ROW = TRUE ,

        firstrow = 1

    ) as x;

create VIEW dbo.Products

AS

SELECT \*

FROM

openrowset(

        BULK N'https://adlsafterhour.dfs.core.windows.net/synapse/PBI\_Course\_Products.csv',

        FORMAT = 'CSV',

        PARSER\_VERSION='2.0',

        HEADER\_ROW = TRUE ,

        firstrow = 1

    ) as x;

create VIEW dbo.Customers

AS

SELECT \*

FROM

openrowset(

        BULK N'https://adlsafterhour.dfs.core.windows.net/synapse/PBI\_Course\_Customers.csv',

        FORMAT = 'CSV',

        PARSER\_VERSION='2.0',

        HEADER\_ROW = TRUE ,

        firstrow = 1

   ) as x;

CREATE  VIEW  dbo.InvoiceDetails

as

SELECT id.\*,i.invoicedate,customerid,invoicetype

FROM

dbo.InvoiceDetails\_temp  as id join dbo.Invoice as i on i.invoiceid = id.invoiceid

-----------------------------------------------

SELECT \* FROM sys.dm\_external\_data\_processed

WHERE type = 'daily'

-----------------------------------------------

create VIEW dbo.Customers\_dwh

AS

SELECT

      [CustomerId] ,

      [FirstName] ,

      [LastName],

      [EmailAddress] ,

      substring(x.filepath(1),CHARINDEX('T',x.filepath(1),CHARINDEX('=',x.filepath(1),1))+1,8) as tempo

FROM

openrowset(

        BULK N'abfss://powerbi@dataflowmp.dfs.core.windows.net/Dataflow/Test/Customers/Customers.csv.snapshots/\*',

        FORMAT = 'CSV',

        firstrow = 1

   )

    WITH(

        [CustomerId] INT,

        [FirstName] VARCHAR(8000),

      [LastName] VARCHAR(8000),

      [EmailAddress] VARCHAR(8000),

      [tempo] VARCHAR(8000)

    )as x;