**Train a simple text classifier**

In text classification scenarios, the goal is to assign a piece of text, such as a document, a news article, a search query, an email, a tweet, support tickets, customer feedback, user product review, to predefined classes or categories. Some examples of text classification applications are: categorizing newspaper articles into topics, organizing web pages into hierarchical categories, spam email filtering, sentiment analysis, predicting user intent from search queries, support tickets routing, and customer feedback analysis.

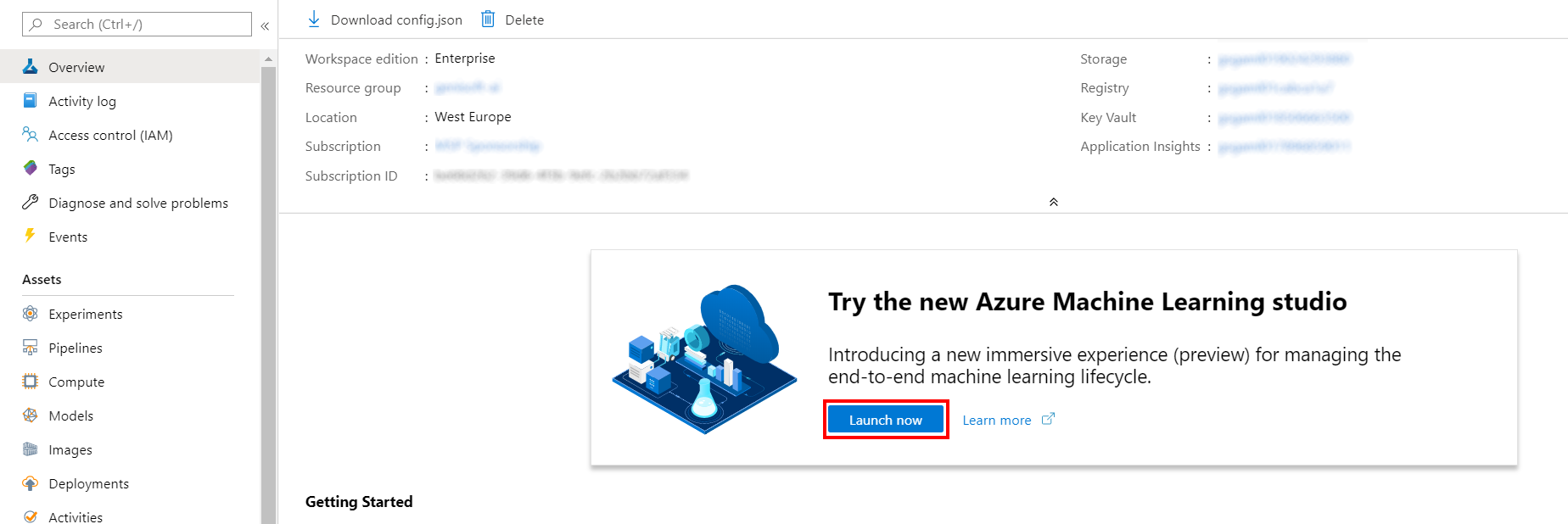
Lab Overview

In this lab we demonstrate how to use text analytics modules available in Azure Machine Learning designer (preview) to build a simple text classification pipeline. We will create a training pipeline and initialize a **multiclass logistic regression classifier** to predict the company category with Wikipedia SP 500 dataset derived from Wikipedia. The dataset manages articles of each S&P 500 company. Before uploading to Azure Machine Learning designer, the dataset was processed as follows: extracted text content for each specific company, removed wiki formatting, removed non-alphanumeric characters, converted all text to lowercase, known company categories added. Articles could not be found for some companies, so that’s why the number of records is less than 500.

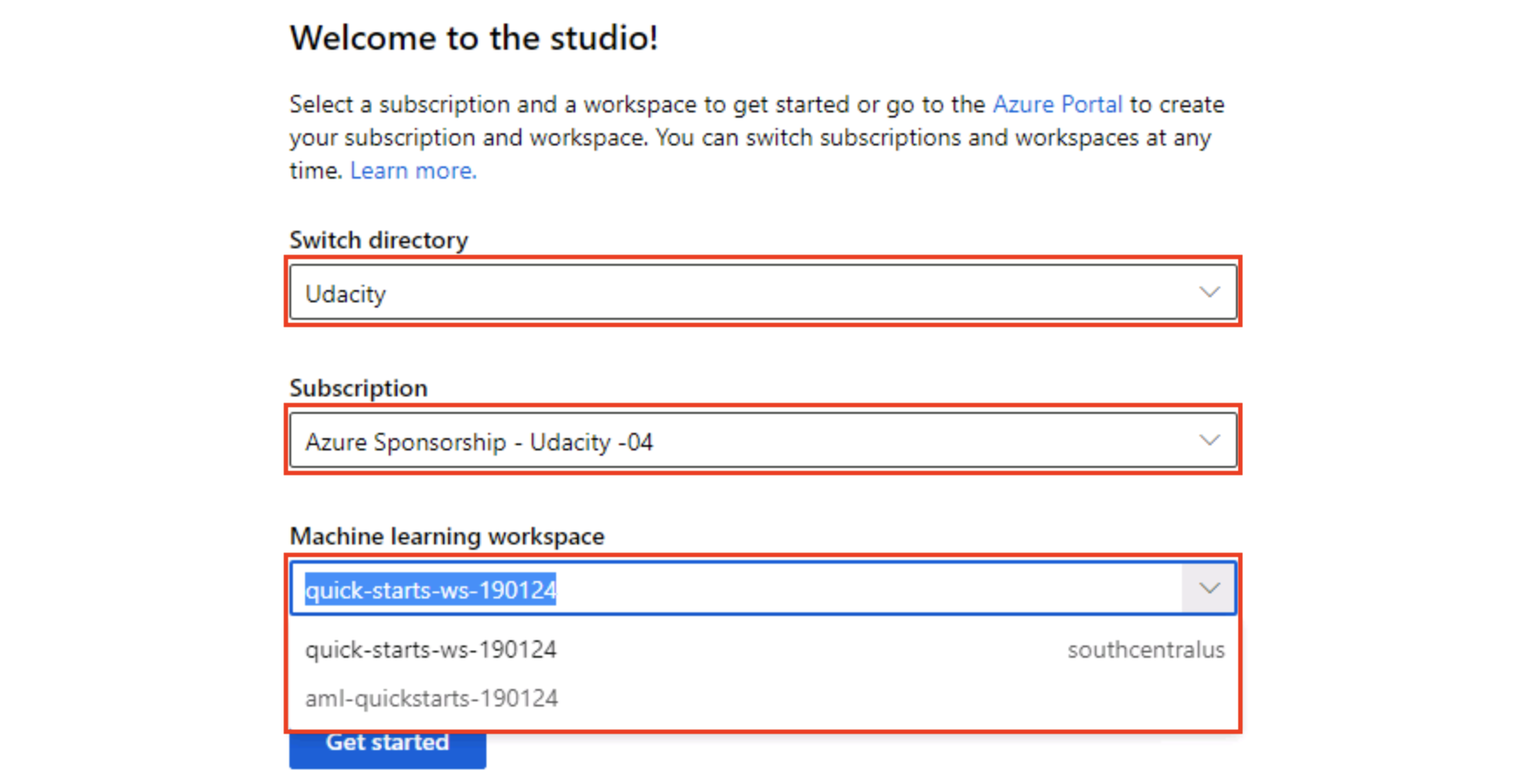
Exercise 1: Create New Training Pipeline

Task 1: Open Pipeline Authoring Editor

1. In [Azure portal](https://portal.azure.com/), open the available machine learning workspace.
2. Select **Launch now** under the **Try the new Azure Machine Learning studio** message.

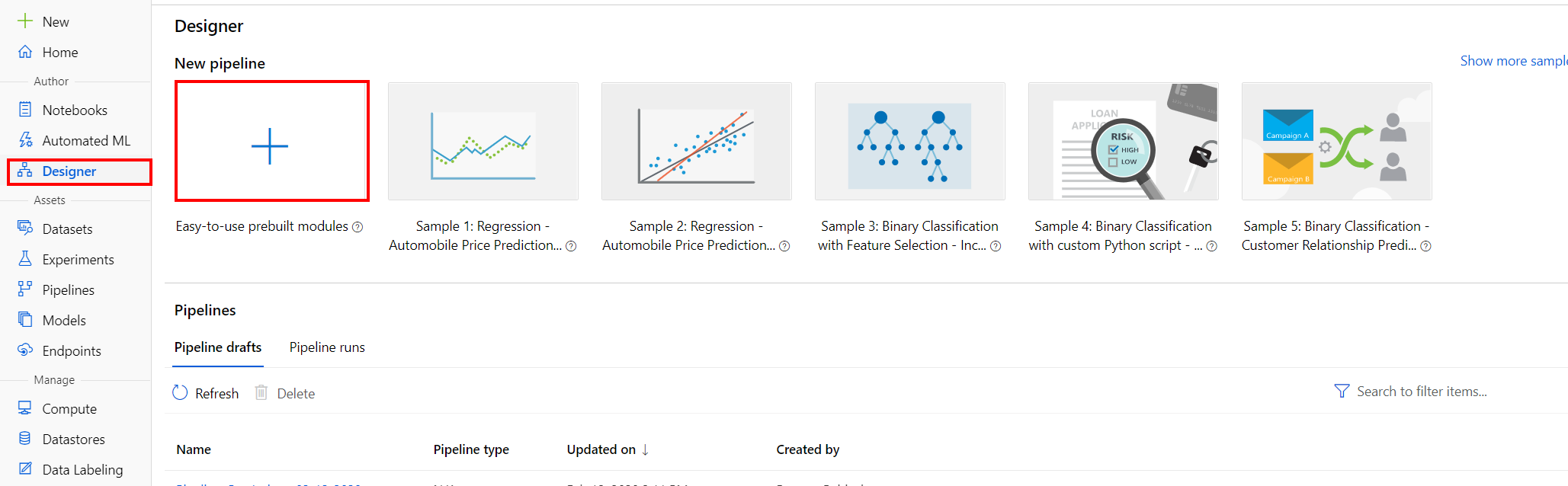


1. When you first launch the studio, you may need to set the directory and subscription. If so, you will see this screen:



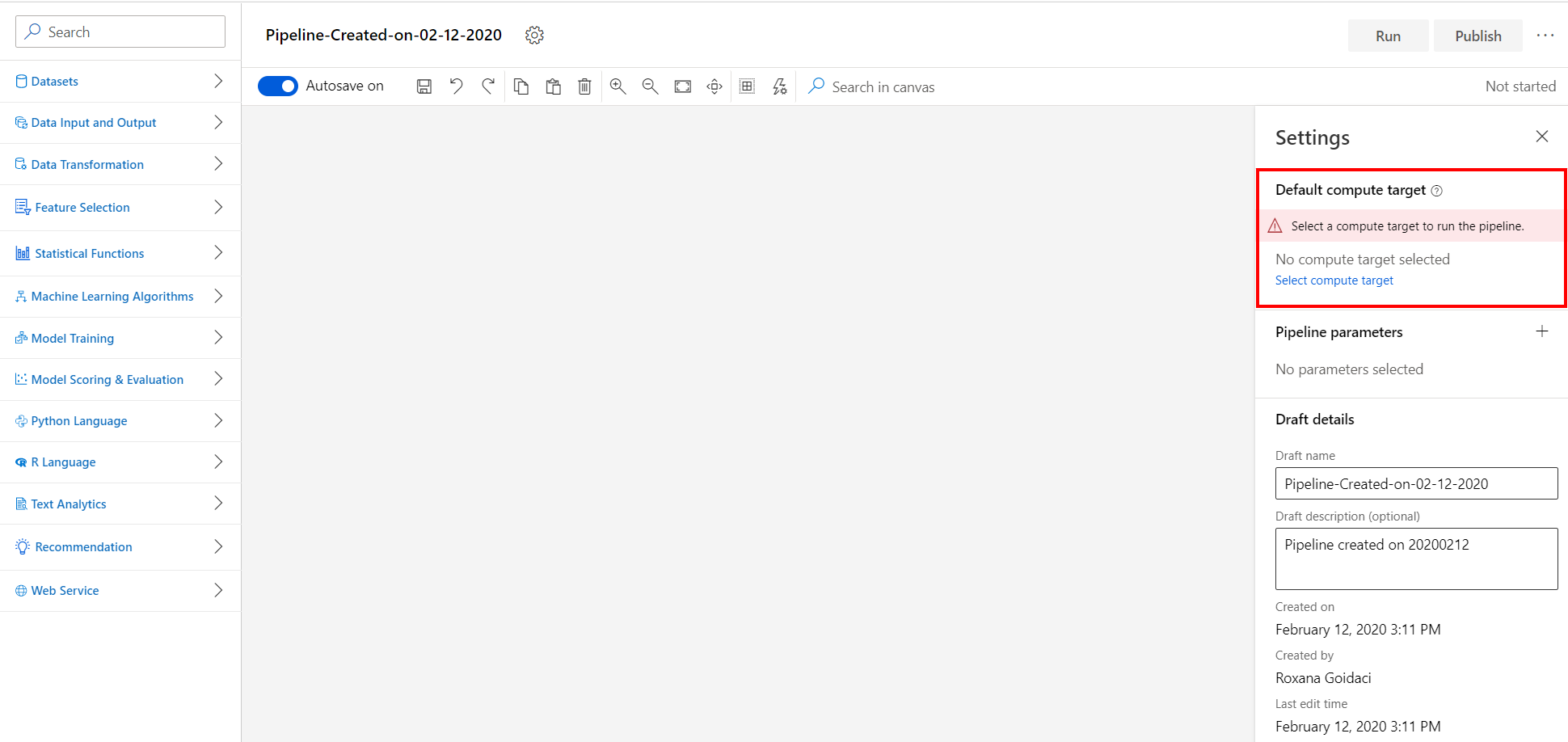
For the directory, select **Udacity** and for the subscription, select **Azure Sponsorship**. For the machine learning workspace, you may see multiple options listed. **Select any of these** (it doesn’t matter which) and then click **Get started**.

1. From the studio, select **Designer, +**. This will open a visual pipeline authoring editor.



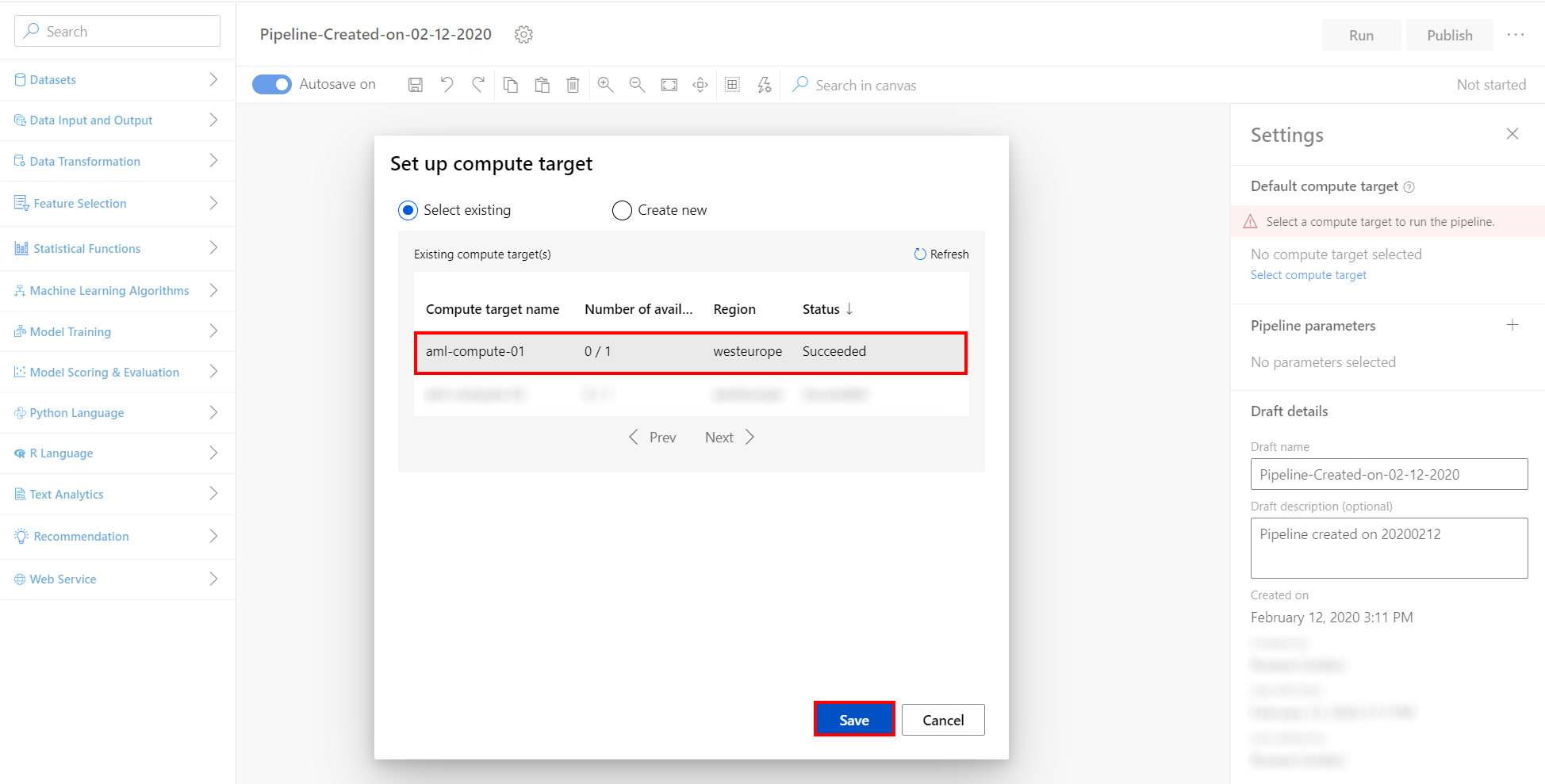
Task 2: Setup Compute Target

1. In the settings panel on the right, select **Select compute target**.



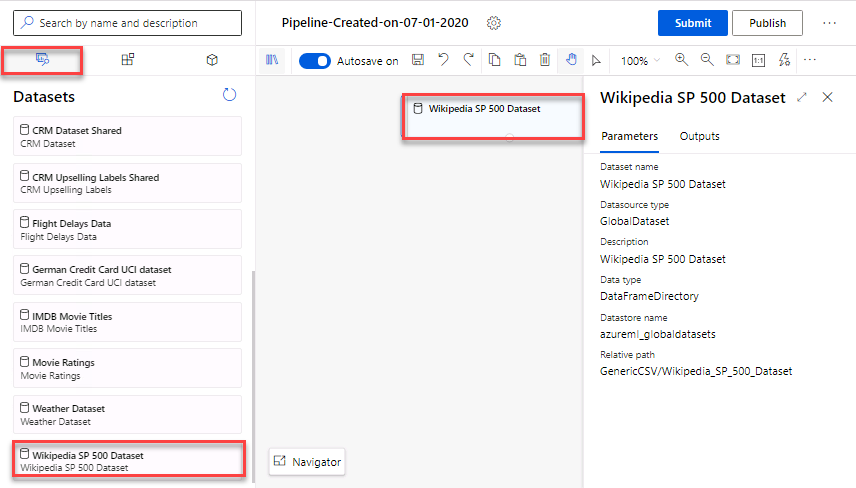
1. In the Set up compute target editor, select the available compute, and then select **Save**.

Note: If you are facing difficulties in accessing pop-up windows or buttons in the user interface, please refer to the Help section in the lab environment.



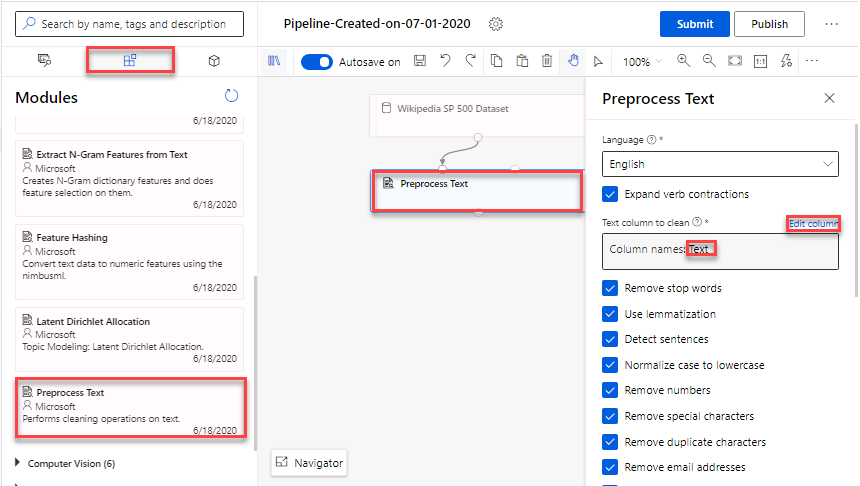
Task 3: Add Wikipedia SP 500 Sample Datasets

1. Select **Datasets** section in the left navigation. Next, select **Samples, Wikipedia SP 500 Dataset** and drag and drop the selected dataset on to the canvas.



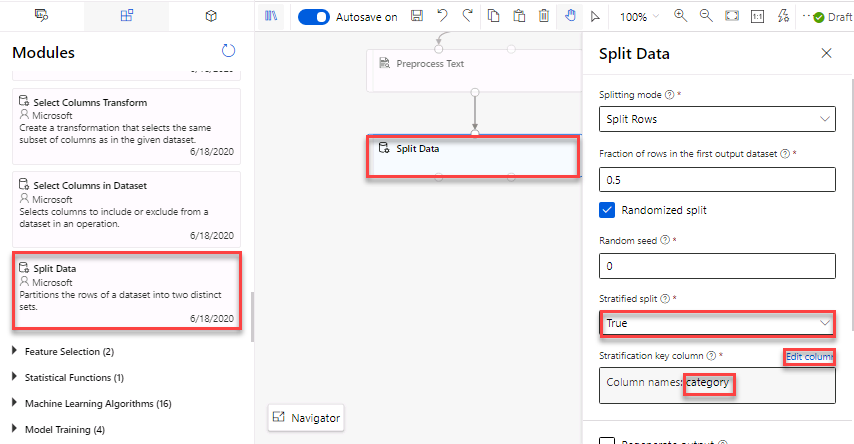
Task 4: Preprocess text for following steps

1. Select **Text Analytics** section in the Modules . Follow the steps outlined below:
   1. Select the **Preprocess Text** prebuilt module.
   2. Drag and drop the selected module on to the canvas.
   3. Connect the output of the Wikipedia SP 500 module to the input of the Preprocess Text module. The dataset input of this prebuilt module needs to be connected to a dataset that has at least one column containing text.
   4. Select the language from the Language dropdown list: **English**.
   5. Select the **Edit column** link to open the Text column to clean editor. Select the **Text** column in the Enter column name field.
   6. Leave all the other options checked, as in the default configuration of the Preprocess Text module.



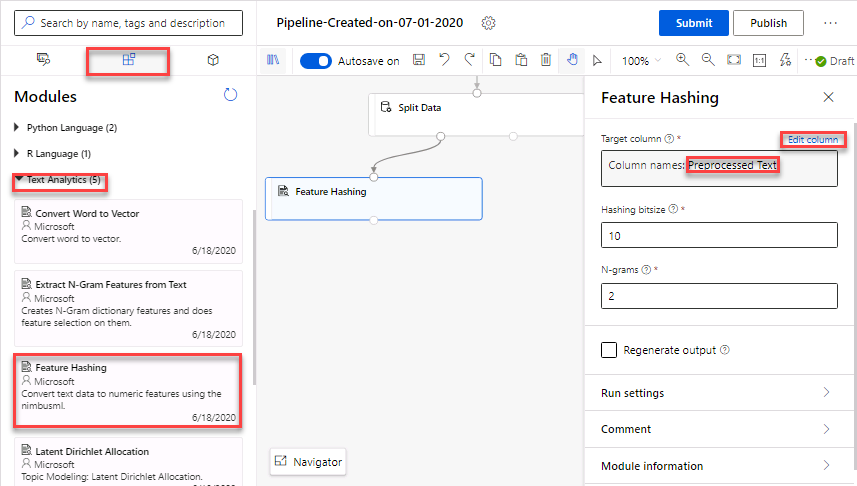
Task 5: Split the dataset into training set (0.5) and test set (0.5)

1. Select **Data Transformation** section in the Modules. Follow the steps outlined below:
   1. Select the **Split Data** prebuilt module.
   2. Drag and drop the selected module on to the canvas.
   3. Fraction of rows in the first output dataset: **0.5**.
   4. Stratified seed: **True**.
   5. Select the **Edit column** link to open the Stratification key column editor. Select the **Category** column in the Enter column name field.
   6. Connect the output of the Preprocess Text module to the input of the Split Data module.



Task 6: Convert the plain text of the articles to integers with Feature Hashing module, on the training set

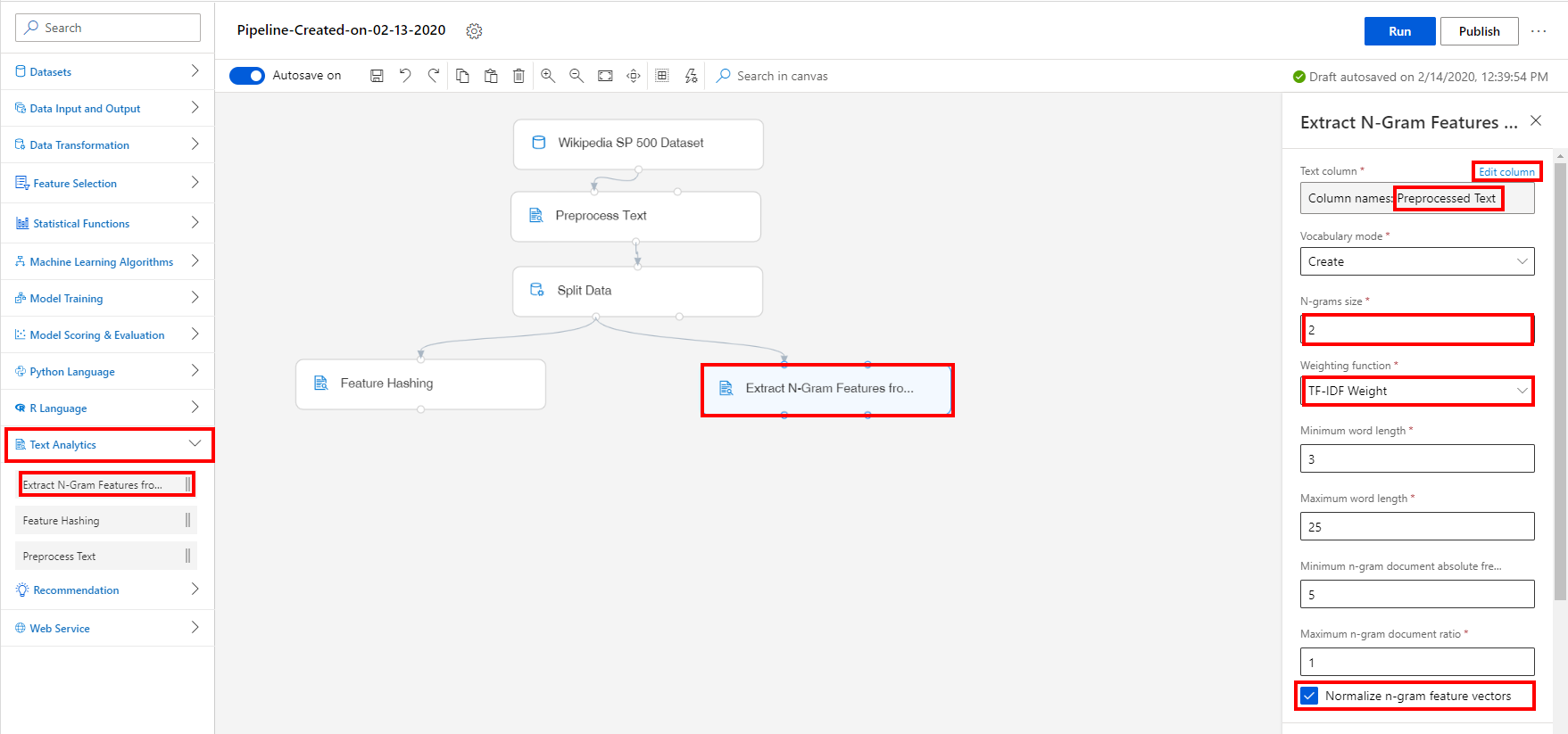
1. Select **Text Analytics** section in the left navigation. Follow the steps outlined below:
   1. Select the **Feature Hashing** prebuilt module.
   2. Drag and drop the selected module on to the canvas.
   3. Connect the first output of the Split Data to the input of the Feature Hashing module.
   4. Select the **Edit columns** link to open the Target column editor and then enter the **Preprocessed Text** column for the Target column field.
   5. Set Hashing bitsize: **10** and set the number of N-grams: **2**.



*The goal of using feature hashing is to reduce dimensionality; also it makes the lookup of feature weights faster at classification time because it uses hash value comparison instead of string comparison.*

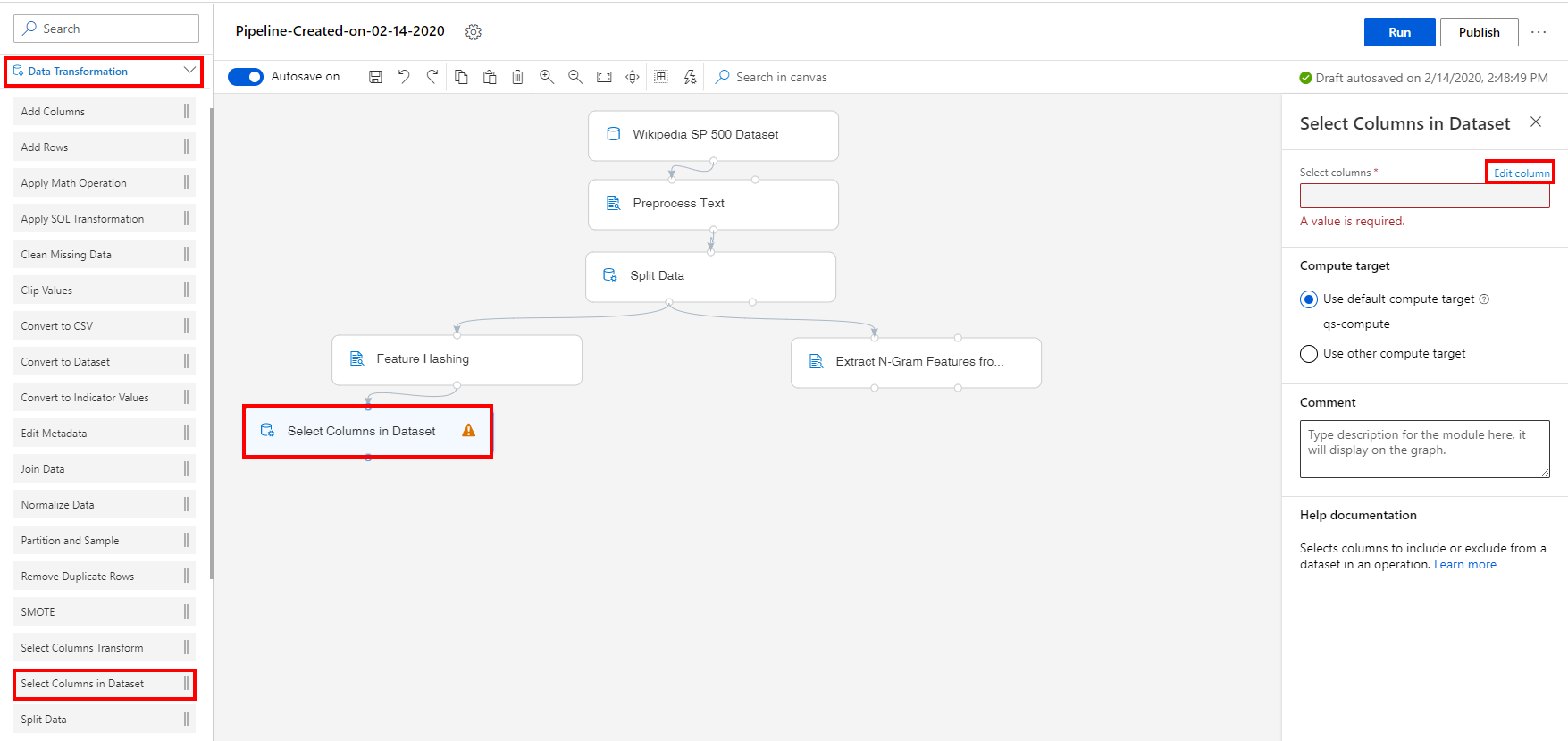
Task 7: Featurize unstructured text data with Extract N-Gram Feature from Text module, on the training set

1. Select **Text Analytics** section in the left navigation. Follow the steps outlined below:
   1. Select the **Extract N-Gram Feature from Text** prebuilt module.
   2. Drag and drop the selected module on to the canvas.
   3. Connect the first output of the Split Data to the input of the Extract N-Gram Feature from Text module.
   4. Select the **Edit columns** link to open the Text column editor and then enter the **Preprocessed Text** column for the Target column field.
   5. Leave default selection of Vocabulary mode to **Create** to indicate that you’re creating a new list of n-gram features.
   6. Set N-grams size: **2** (which is the maximum size of the n-grams to extract and store)
   7. Select Weighting function: **TF-IDF Weight**. (This function calculates a term frequency/inverse document frequency score and assigns it to each n-gram. The value for each n-gram is its TF score multiplied by its IDF score.)
   8. Check the option to **Normalize n-gram feature vectors**. If this option is enabled, each n-gram feature vector is divided by its L2 norm.

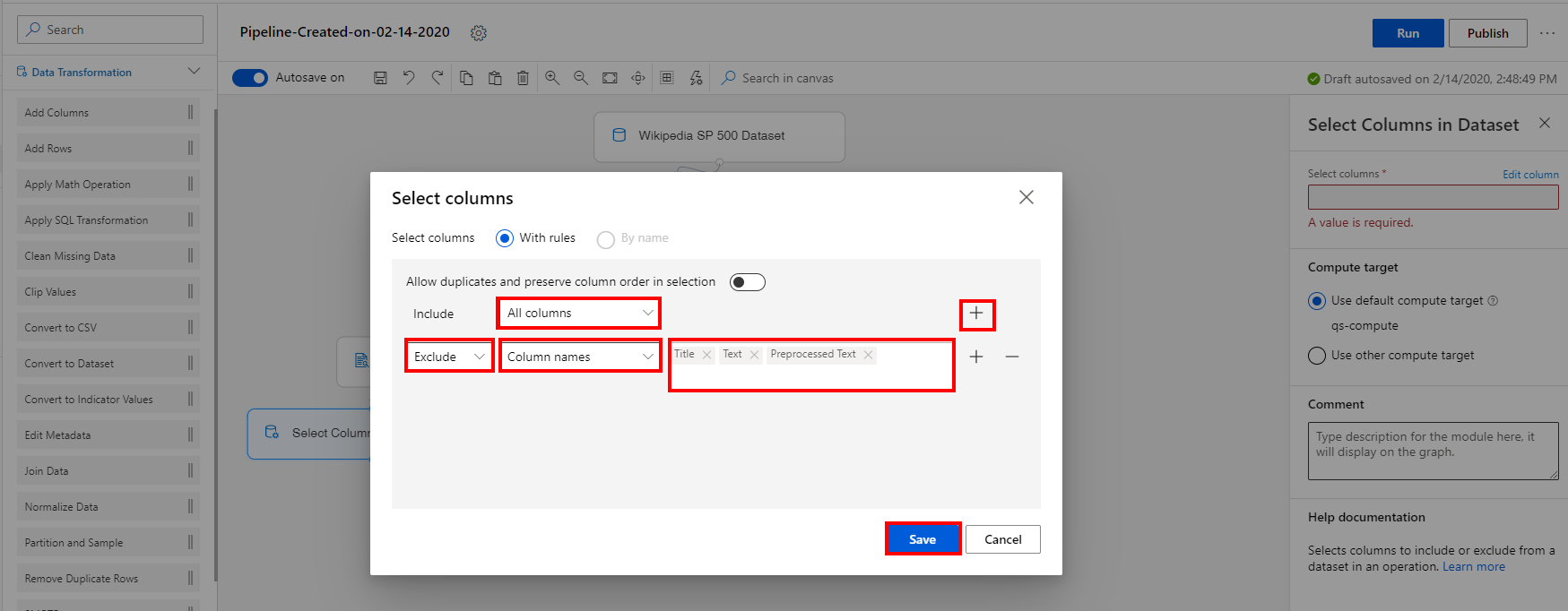


Task 8: Remove text columns from dataset

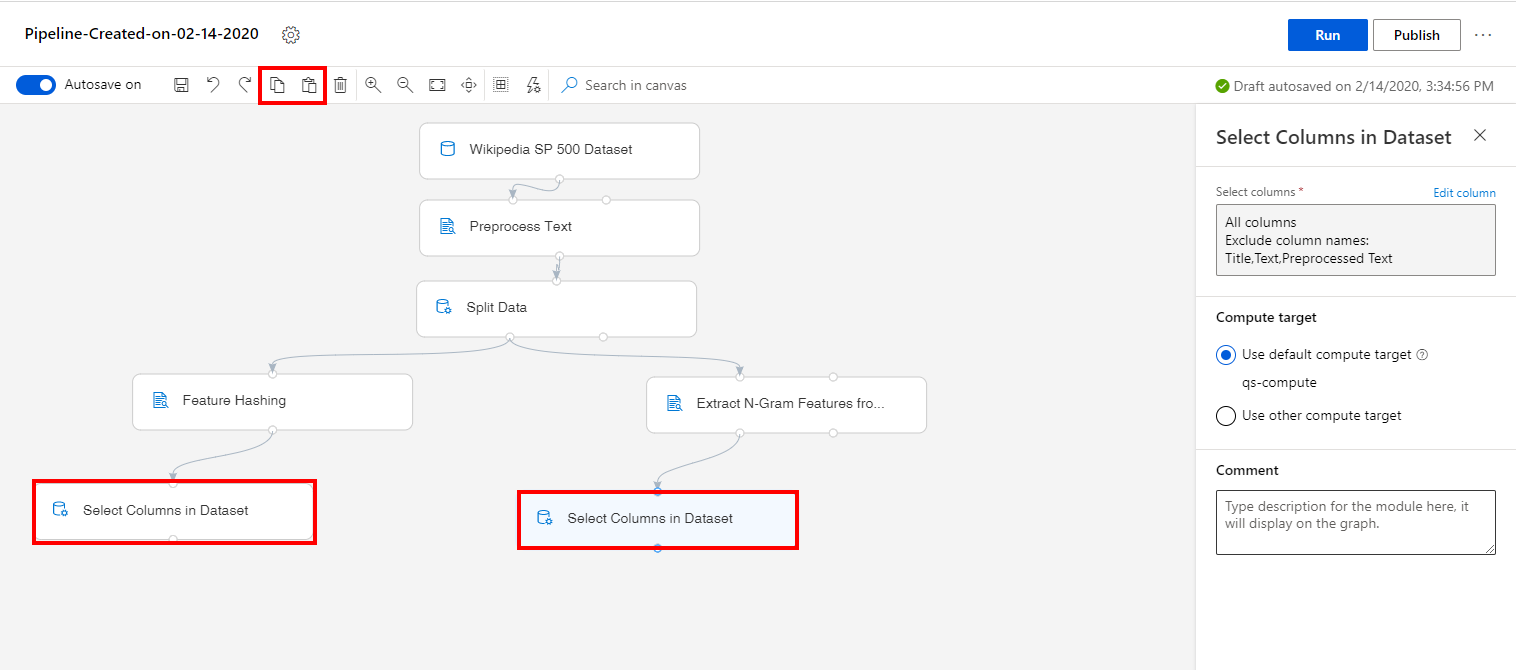
1. Select **Data Transformation** section in the left navigation. Follow the steps outlined below to add two Select Columns in Dataset modules on both featurization branches:
   1. Select the **Select Columns in Dataset** prebuilt module
   2. Drag and drop the selected module on to the canvas
   3. Connect the input Select Columns in Dataset module to the Feature Hashing module output.



1. Select the **Edit columns** link to open the Select columns editor and then select **All columns**.
2. Click on the **+** icon to add another operation line and select the **Exclude, Column names** option. Enter to following list of columns to be excluded: **Title**, **Text**, **Preprocessed Text**. Select **Save** to close the editor.

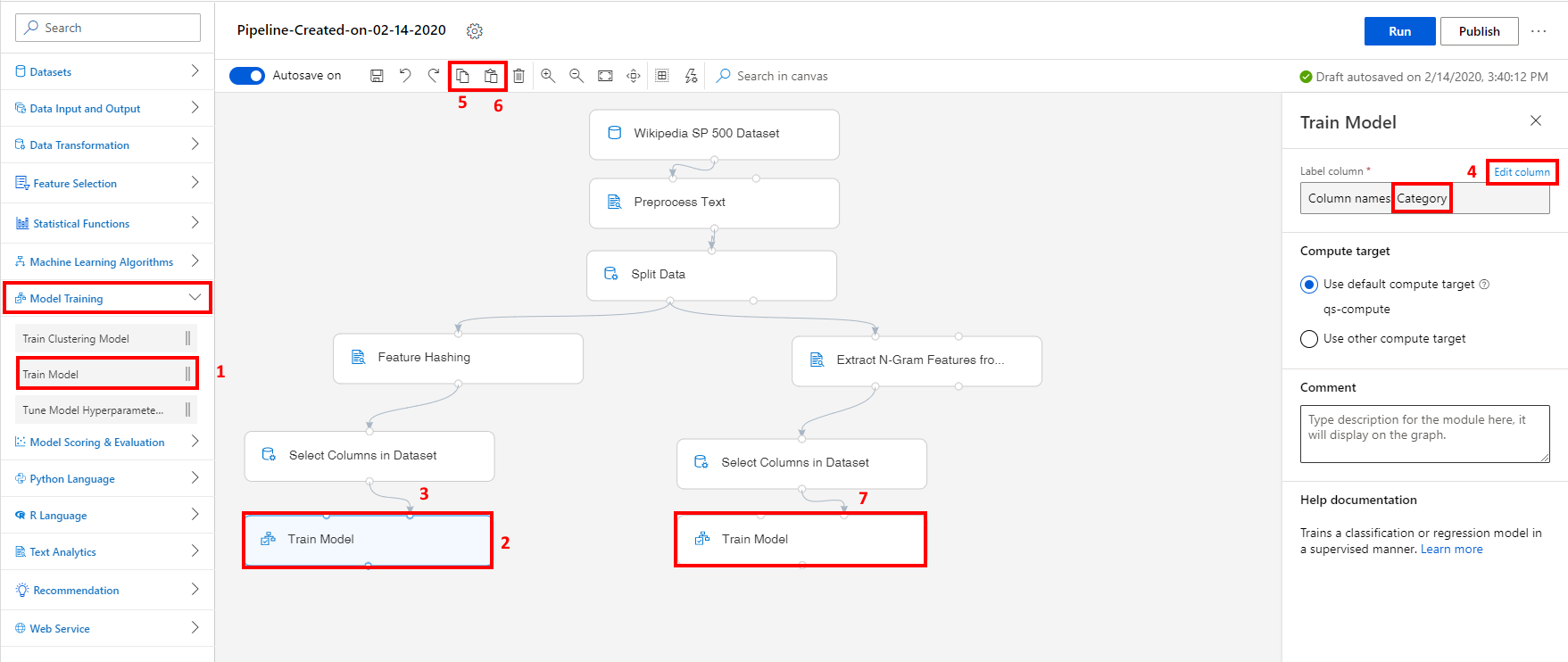


1. Select the Select Columns in Dataset module and click on the **Copy** icon and then **Paste** it on the canvas. Connect the copied module to the Extract N-Gram Feature from Text module.



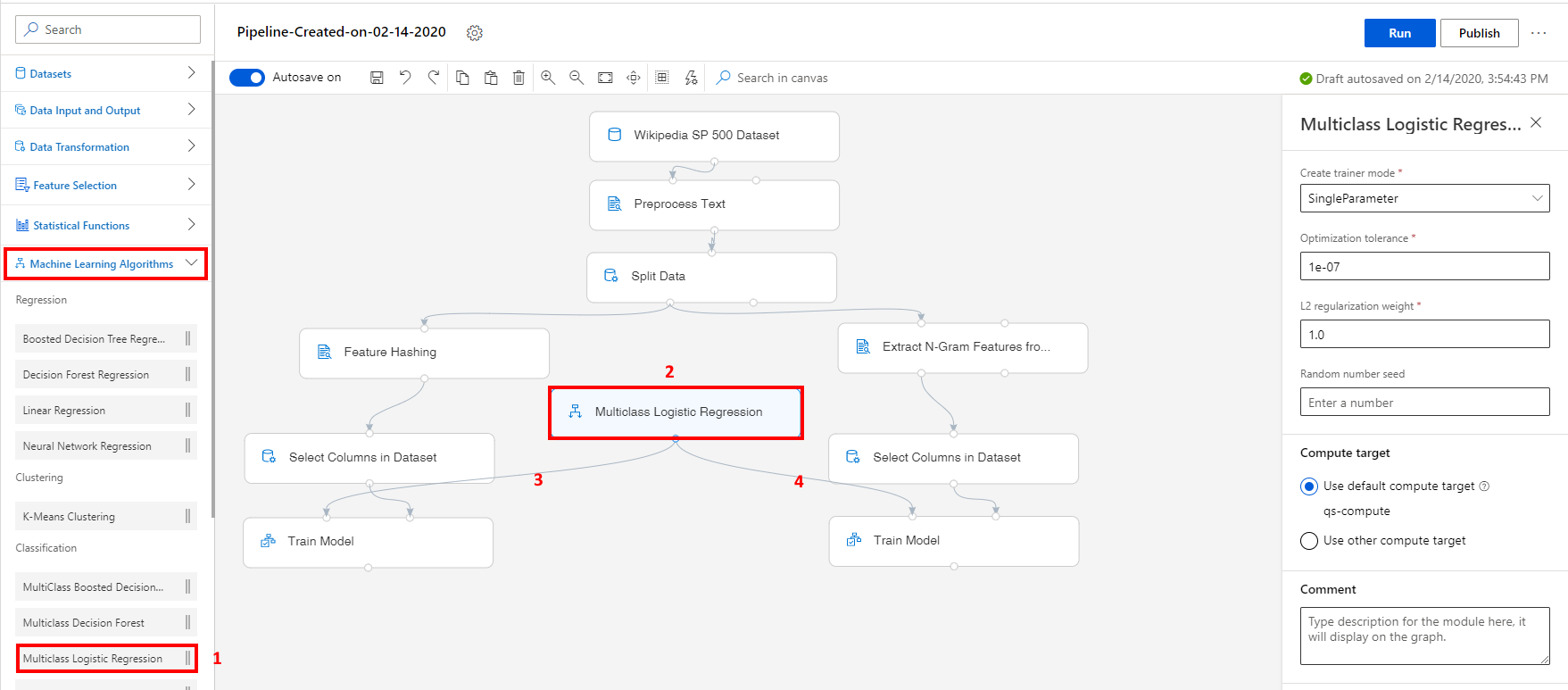
Task 9: Add the Train Model Modules

1. Select **Model Training** section in the left navigation. Follow the steps outlined below:
   1. Select the **Train Model** prebuilt module.
   2. Drag and drop the selected module on to the canvas
   3. Connect the first output of the Select Columns in Dataset module on the left of the canvas, to the second input (the Dataset input) of the newly added Train Model module.
   4. Select the **Edit columns** link to open the Label column editor and then enter the **Category** column. Select **Save** to close the editor.
   5. Select the Train Model module and click on the **Copy** icon.
   6. Click on the **Paste** icon to paste a second Train Model module on the canvas under the right branch of the pipeline tree.
   7. Connect this last Train Model module, by linking at the second input (the Dataset input), the output of the right most Select Columns in Dataset module on the canvas.



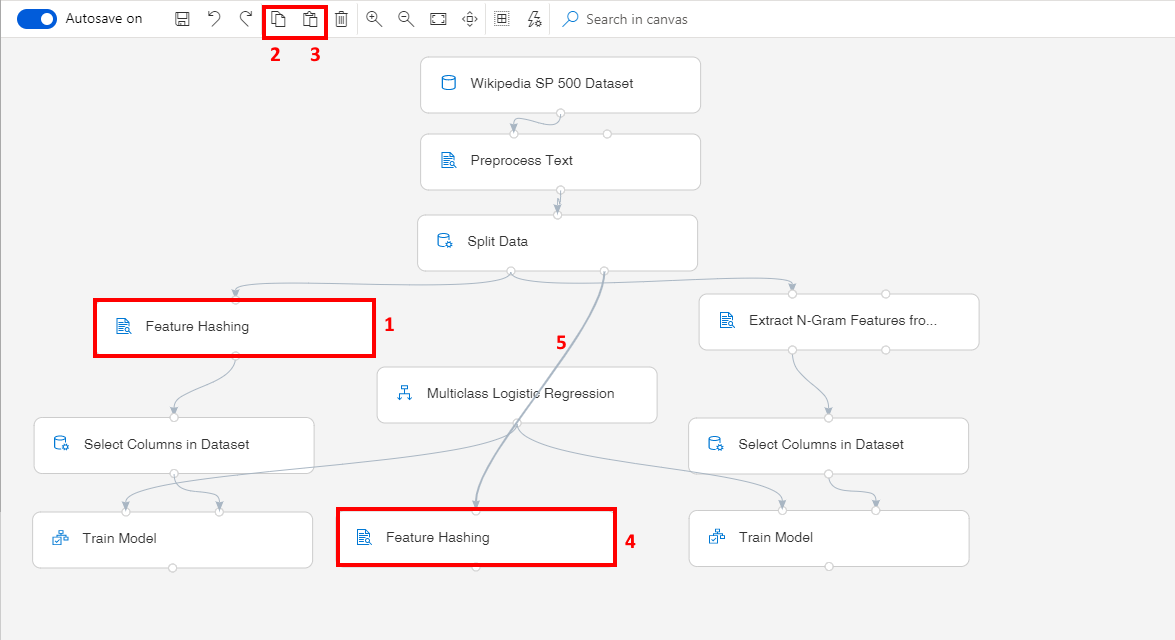
Task 10: Initialize Multiclass Logistic Regression Model

1. Select **Machine Learning Algorithms** section in the left navigation. Follow the steps outlined below:
   1. Select the **Multiclass Logistic Regression** prebuilt module, in the Classification category.
   2. Drag and drop the selected module on to the canvas
   3. Connect the output of the Multiclass Logistic Regression module to the first input of the left branch (the Feature Hashing approach) Train Model module.
   4. Connect the output of the Multiclass Logistic Regression module to the first input of the right branch (the N-Gram Features approach) Train Model module.



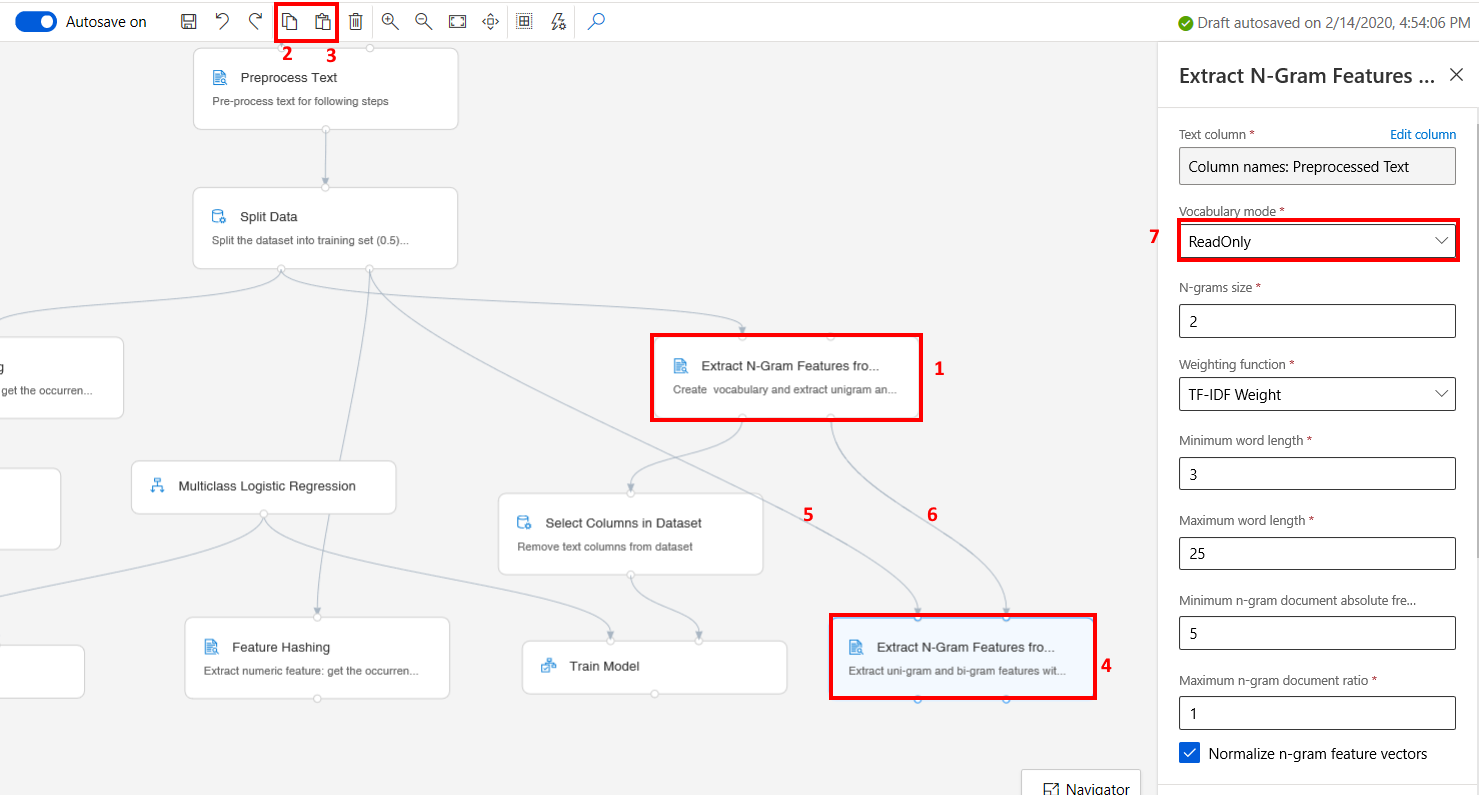
Task 11: Convert the plain text of the articles to integers with Feature Hashing module, on the test set

1. Select the existing **Feature Hashing** module.
2. Copy the **Feature Hashing** module using the **Copy** option from the top menu.
3. **Paste** the copied module on the canvas.
4. Position the new **Feature Hashing** module between the two Train Model modules on the canvas.
5. Connect the second output of the Split Data, the test set output, to the input of the newly added Feature Hashing module.



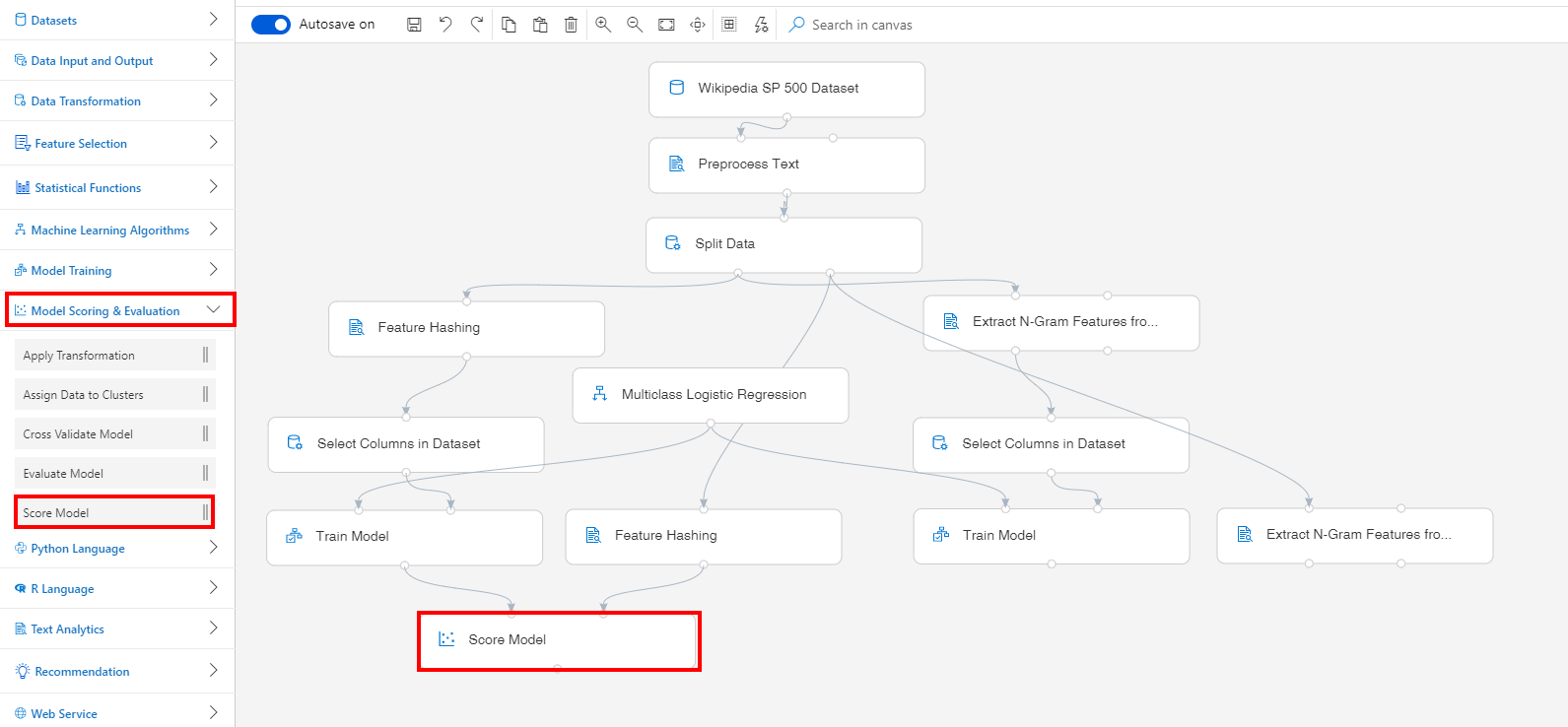
Task 12: Featurize unstructured text data with Extract N-Gram Feature from Text module, on the test set

1. Select the existing **Extract N-Gram Feature from Text** module.
2. Copy the **Extract N-Gram Feature from Text** module using the **Copy** option from the top menu.
3. **Paste** the copied module on the canvas.
4. Position the new **Extract N-Gram Feature from Text** module near the rightmost Train Model module on the canvas.
5. Connect the second output of the Split Data, the test set output, to the first input of the newly added Extract N-Gram Feature from Text module.
6. Connect the second output of the uppermost Extract N-Gram Feature from Text, to the second input of the copied Extract N-Gram Feature from Text module.
7. Select the newly added module and in the right settings pane, set the Vocabulary mode to **ReadOnly**

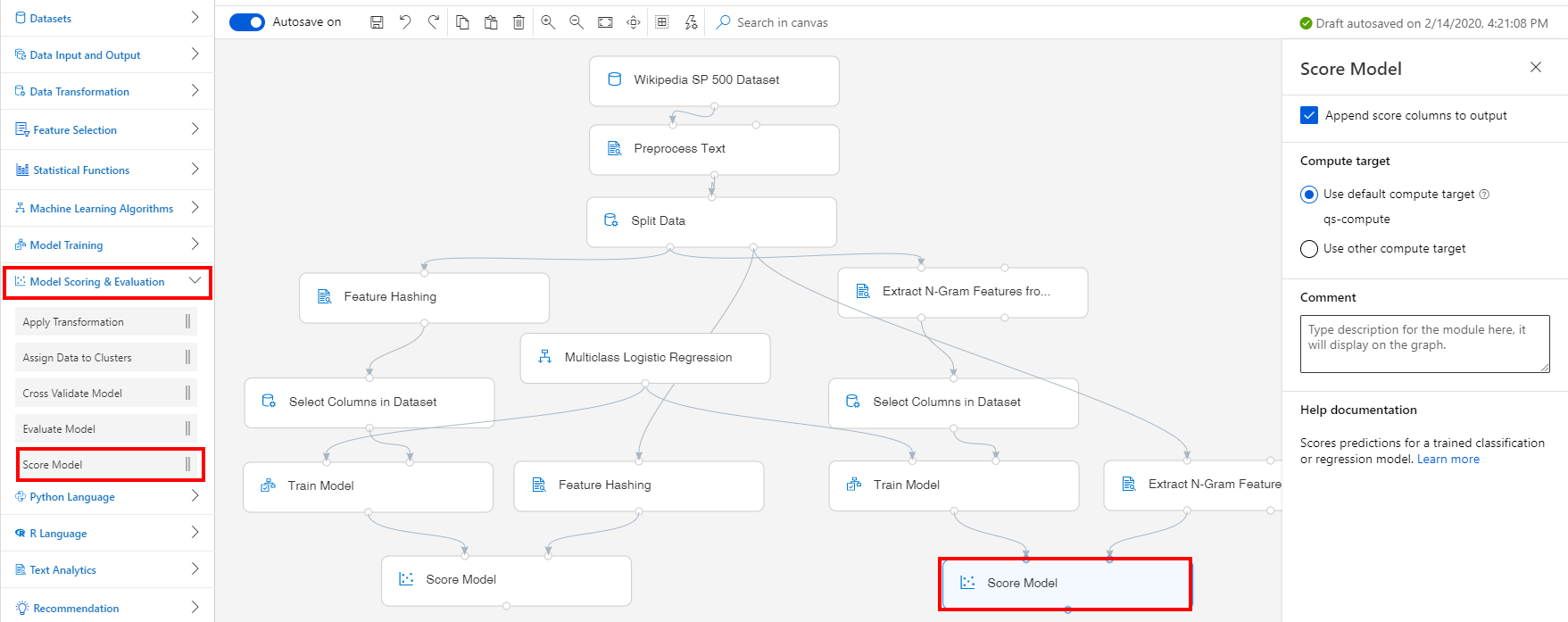


Task 13: Setup Score Model Modules

1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Score Model** prebuilt module
   2. Drag and drop the selected module on to the canvas
   3. Connect the leftmost Train Model module to the first input of the Score Model module
   4. Connect the output of the Feature Hashing module (the lower one, on the test set branch) to the second input of the Score Model module

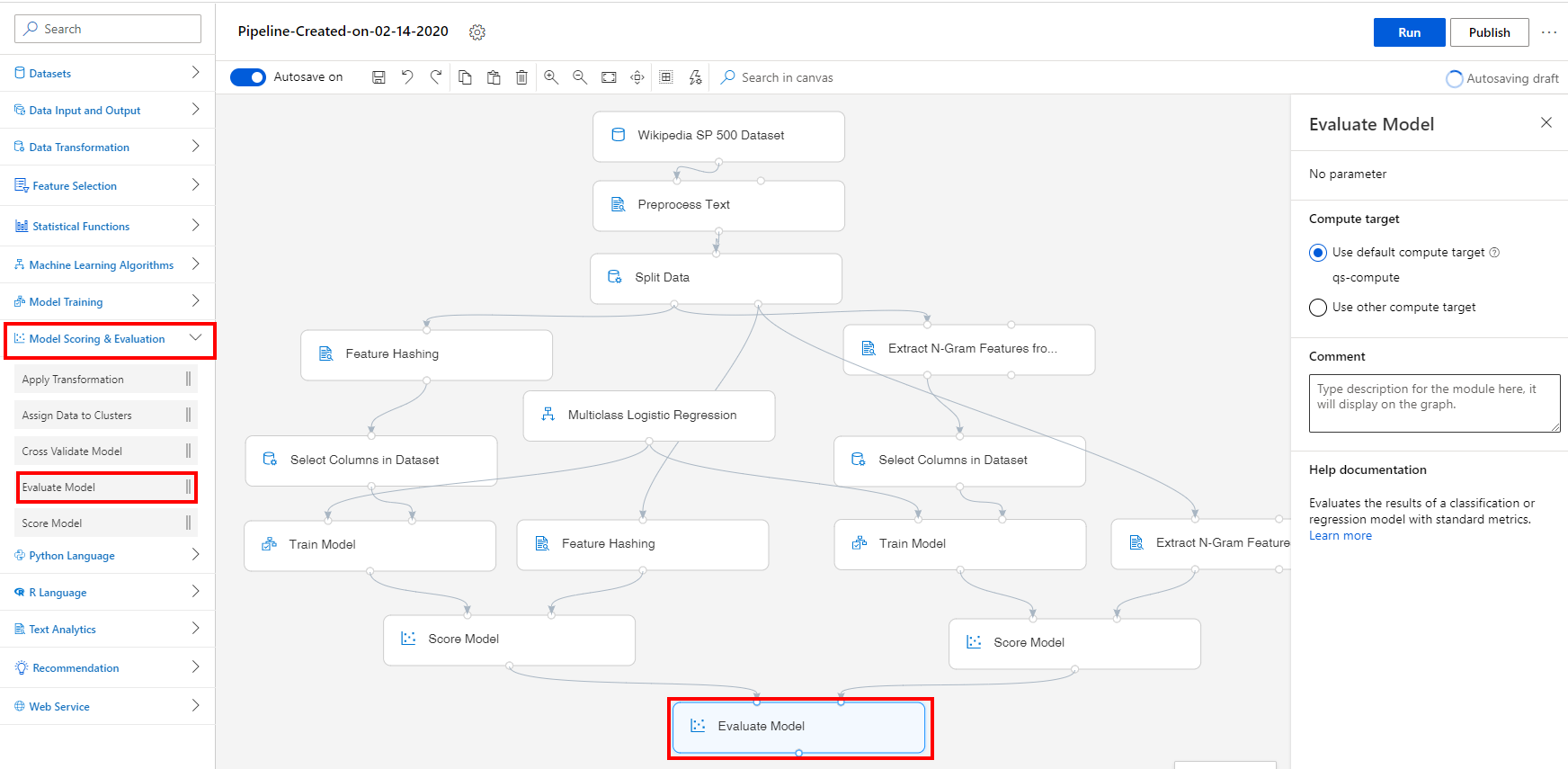


1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Score Model** prebuilt module
   2. Drag and drop the selected module on to the canvas
   3. Connect the rightmost Train Model module to the first input of the Score Model module
   4. Connect the first output of the Extract N-Gram Feature from Text module (the lower one, on the test set branch) to the second input of the rightmost Score Model module.



Task 14: Setup Evaluate Model Module

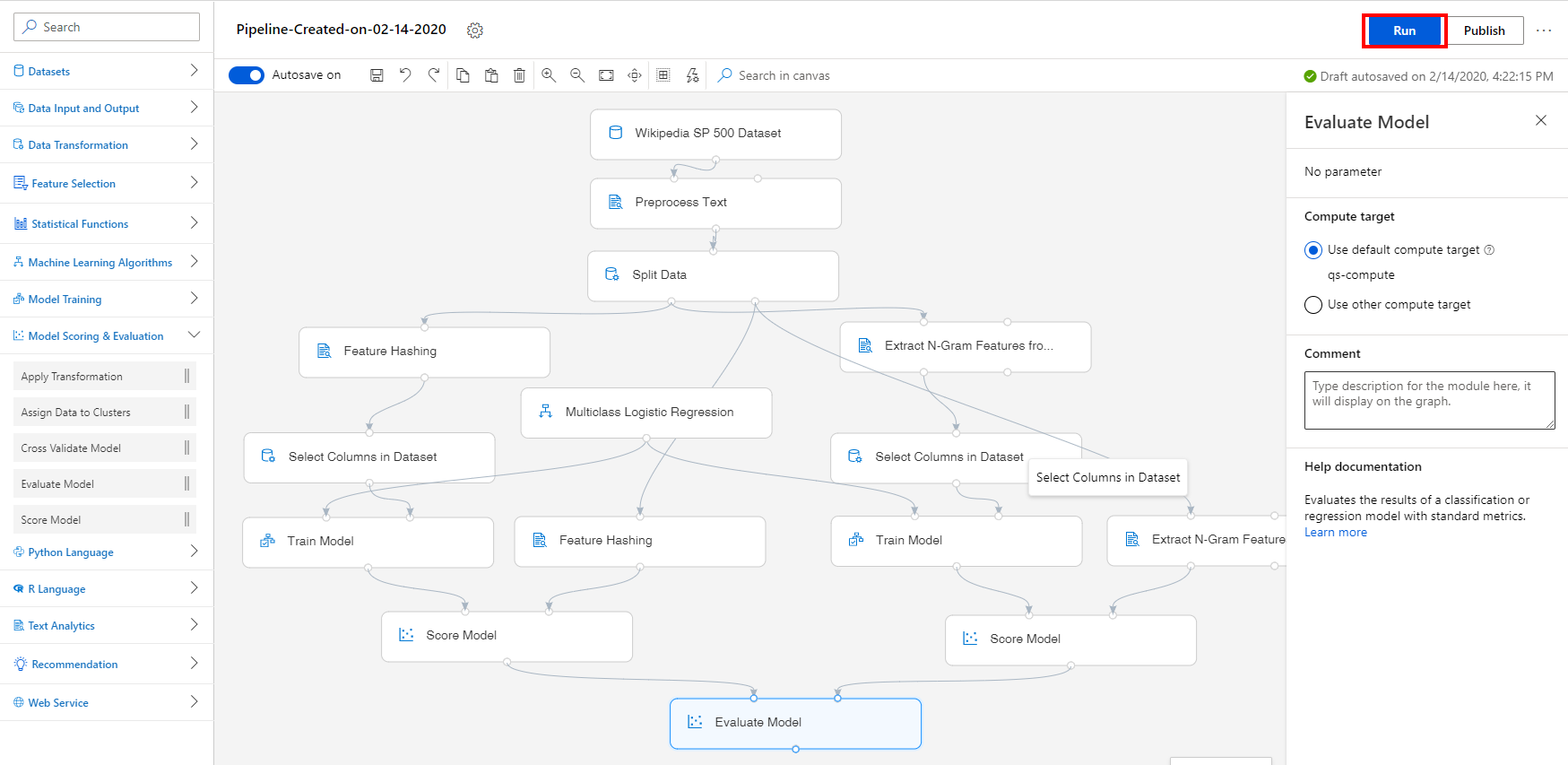
1. Select **Model Scoring & Evaluation** section in the left navigation. Follow the steps outlined below:
   1. Select the **Evaluate Model** prebuilt module
   2. Drag and drop the selected module on to the canvas
   3. Connect the two Score Model modules to the inputs of the Evaluate Model module



Exercise 2: Submit Training Pipeline

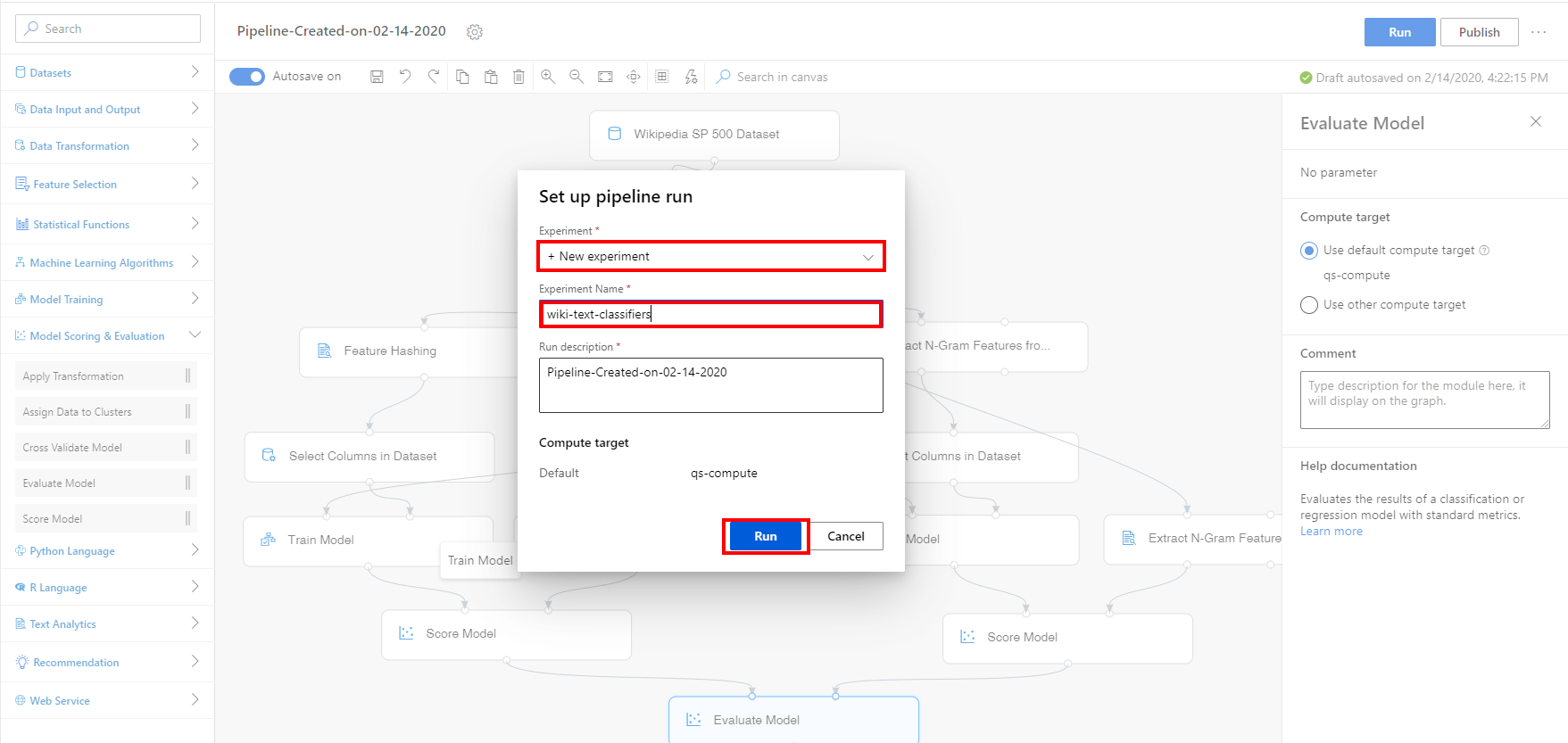
Task 1: Create Experiment and Submit Pipeline

1. Select **Submit** to open the Setup pipeline run editor.



Please note that the button name in the UI is changed from **Run** to **Submit**.

1. In the Setup pipeline run editor, select **Experiment, Create new** and provide New experiment name: **wiki-text-classifier**, and then select **Submit**.

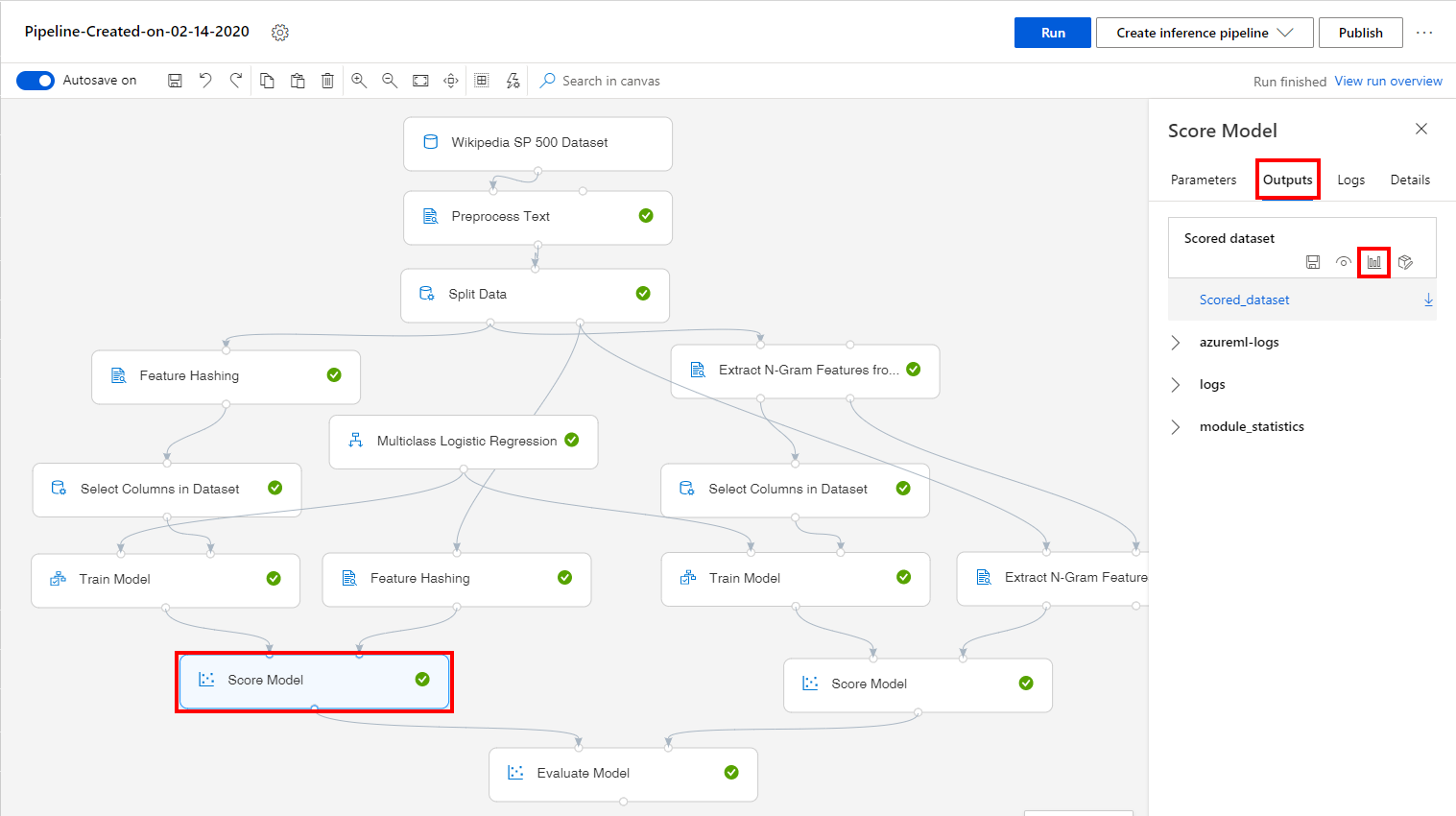


1. Wait for pipeline run to complete. It will take around **20 minutes** to complete the run.
2. While you wait for the model training to complete, you can learn more about the classification algorithm used in this lab by selecting [Multiclass Logistic Regression module](https://docs.microsoft.com/en-us/azure/machine-learning/algorithm-module-reference/multiclass-logistic-regression).

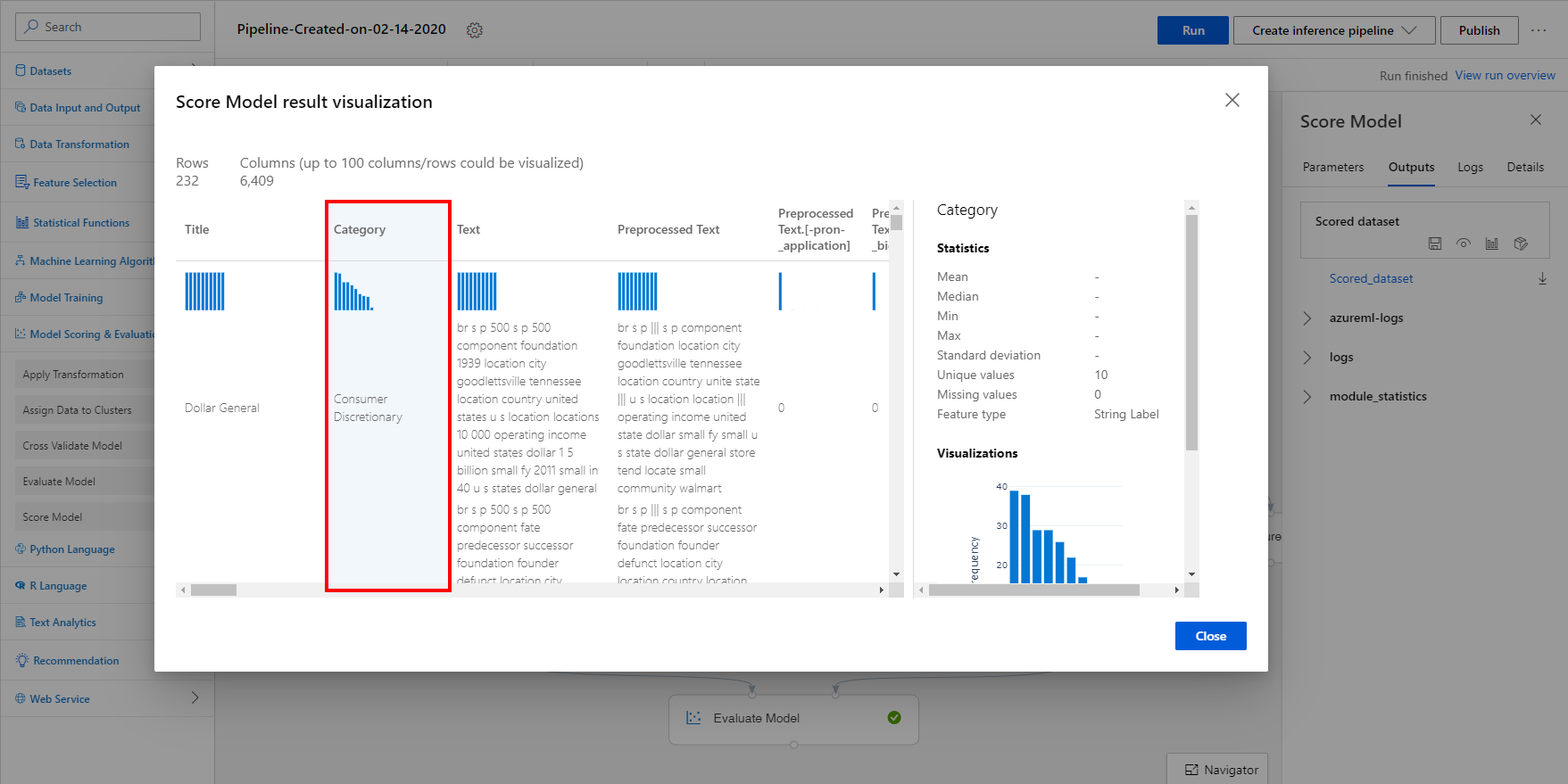
Exercise 3: Visualize Training Results

Task 1: Visualize the Model Predictions

1. Select **Score Model, Outputs, Visualize** to open the Score Model result visualization dialog or just simply right-click the Score Model module and select **Visualize Scored Dataset**.

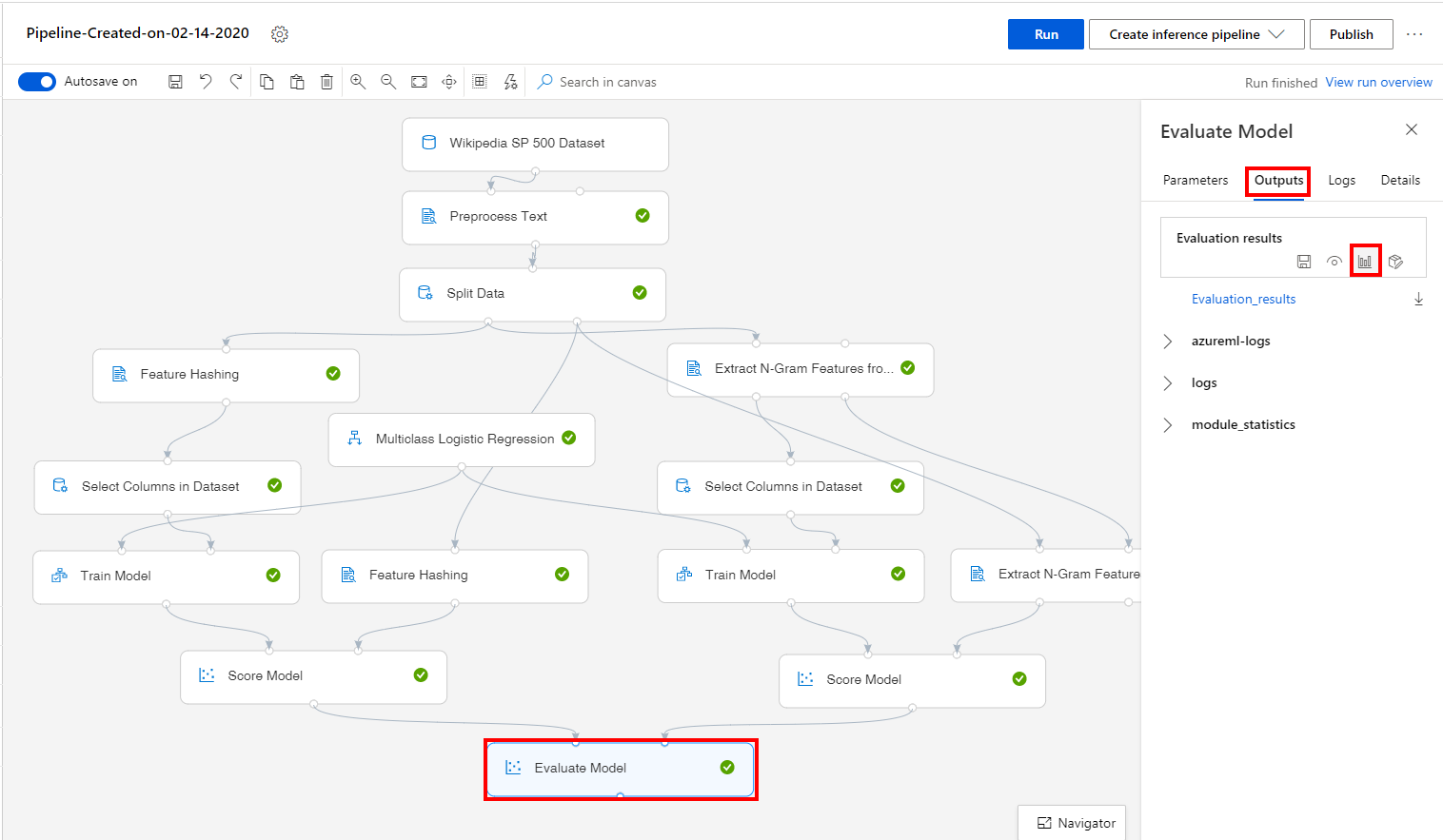


1. Observe the predicted values under the column **Category**.

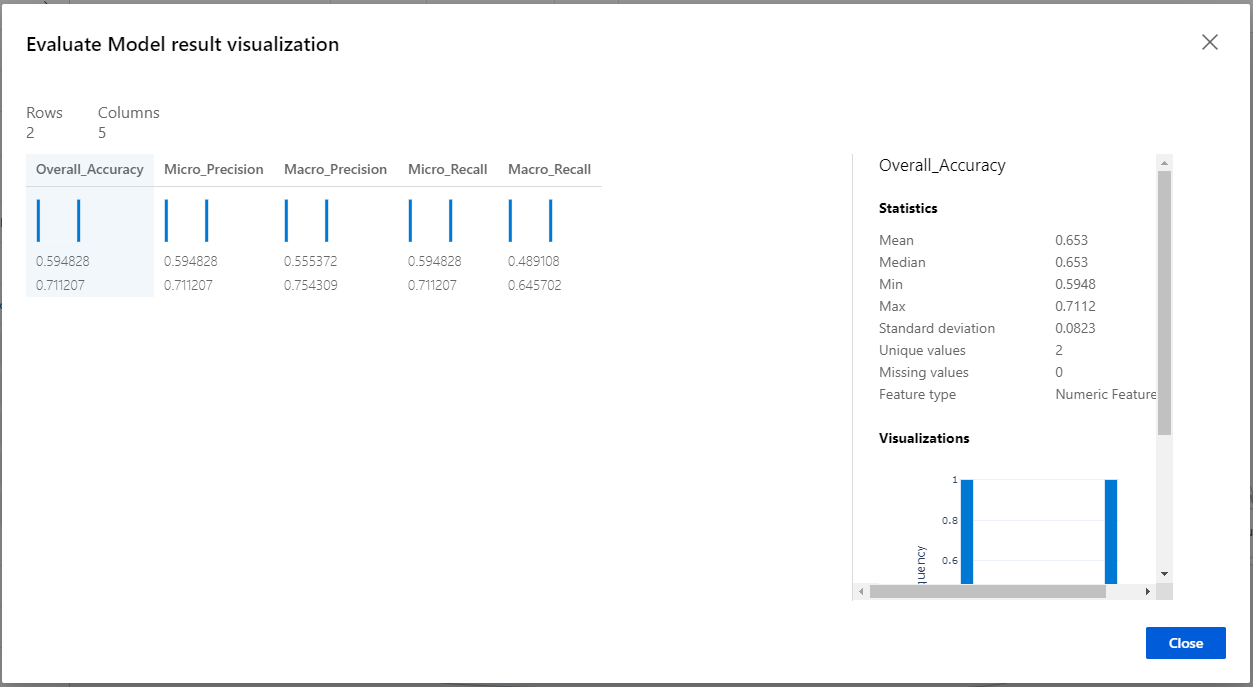


Task 2: Visualize the Evaluation Results

1. Select **Evaluate Model, Outputs, Visualize** to open the Evaluate Model result visualization dialog or just simply right-click the Evaluate Model module and select **Visualize Evaluation Results**.



1. Evaluate the model performance by reviewing the various evaluation metrics. Evaluate Model has two input ports, so that we could evaluate and compare scored datasets that are generated with different methods. In this sample, we compare the performance of the result generated with feature hashing method and n-gram method.



Next Steps

Congratulations! You have trained a simple text classifier and compared performance of the result generated with two different featurization modules. You can continue to experiment in the environment but are free to close the lab environment tab and return to the Udacity portal to continue with the lesson.