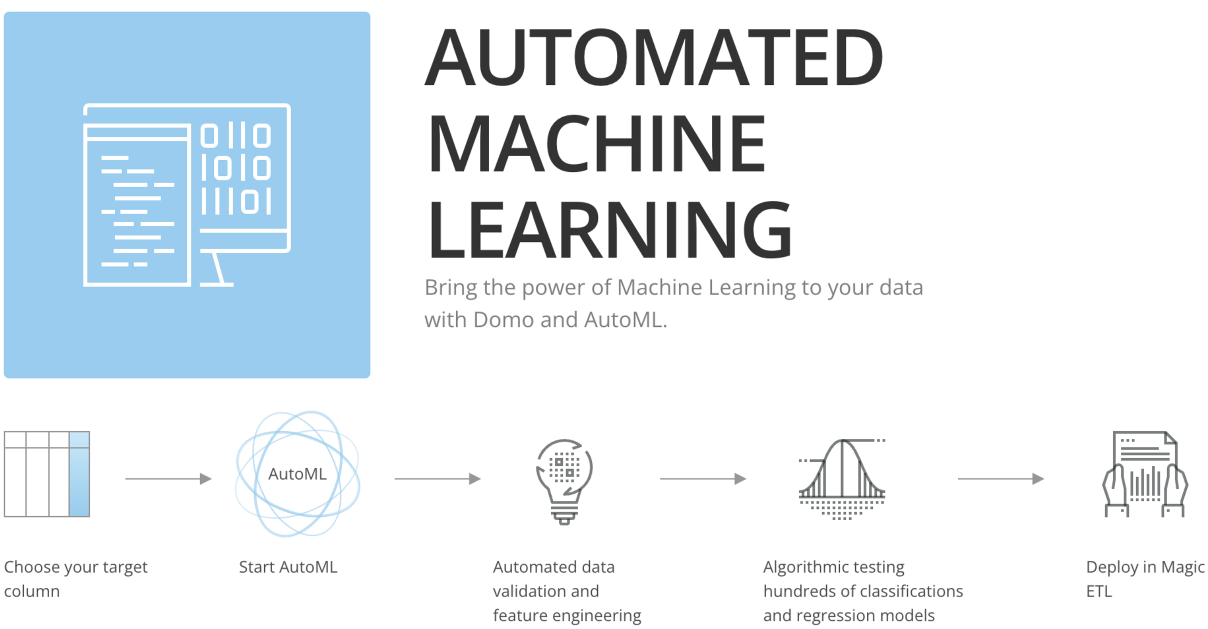
AutoML [Automated Machine Learning]

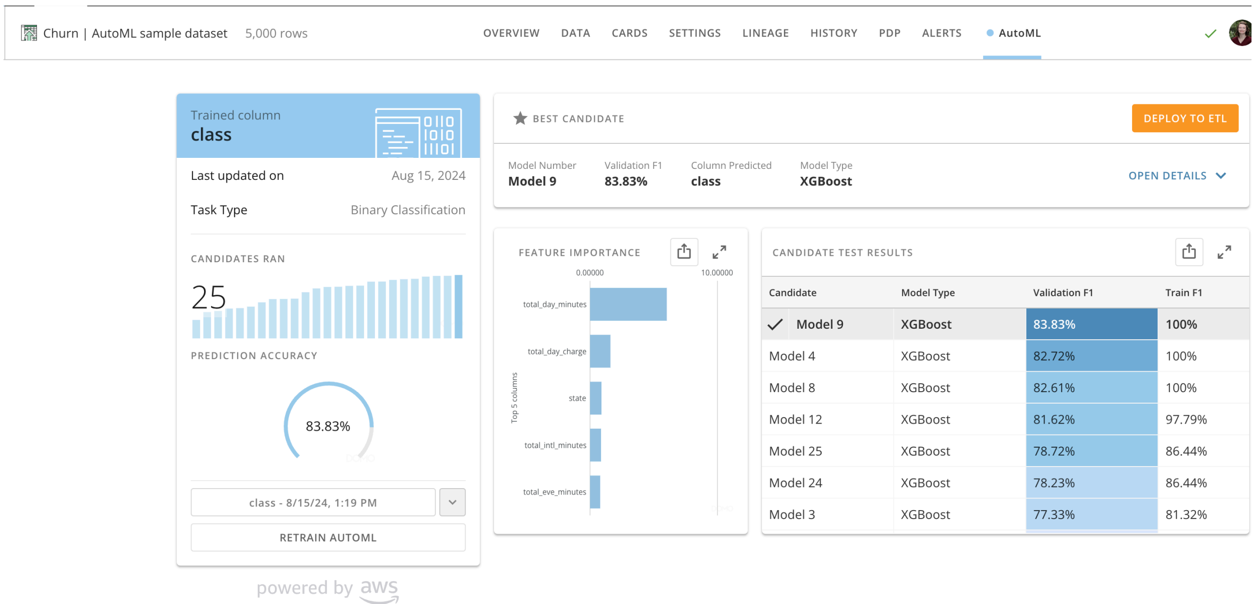
Intro

powered by aws

Machine learning is within everyone's grasp with Domo's automated machine learning [AutoML] tool. In partnership with Amazon SageMaker Autopilot, AutoML automatically trains and tunes machine learning models based on data provided by a customer. Specifically, with just a few clicks, AutoML will transform your data to be ready for machine learning and launch hundreds of training jobs on any DataSet in Domo to find the model that achieves the best performance for your task. You can then easily deploy the model on your Domo DataSets with the AutoML Interface tile in Magic ETL.

Check out [this video] to see it in action.





- Required Grants
- Enable AutoML
- Prepare Data for AutoML
- Using AutoML
  - Launch an AutoML Training Job
  - Generate Predictions Using the AutoML Inference Tile in Magic ETL

Required Grants

To access AutoML, you need the Execute AutoML grant enabled. You can add this grant to a custom Domo role.

- **Execute AutoML —** Train AutoML models. Run DataFlows containing AutoML Inference actions.

Enable AutoML

AutoML is available by default for users on the Domo Consumption agreement.

For non-consumption users, it is available *on demand* and *paid*. To enable AutoML, contact your Domo account team.

Prepare Data for AutoML

Having a well-prepared and cleaned DataSet is critical for successful outcomes when using the AutoML tool. Below, we provide a data prep checklist for AutoML.

- **Data Structure:** The following are guidelines for structuring your DataSet:
  - **Singular DataSet:** Spend some time gathering your data together into one DataSet. Your DataSet should include both your output variable (the variable you want to predict) and your input variables (variables you will use to predict and that you expect to have an effect/influence on your output variable). Input variables are also commonly referred to as “features” by machine learning practitioners.
  - **Identify the outcome:** Be sure to only include one output column for your DataSet.
  - **Unit of analysis:** Each row should encapsulate one record in your business problem. For example, if you would like to predict which sales opportunity will close, each row should enumerate one distinct sales opportunity, from start to finish, with the output column listing the result of the sales opportunity (i.e. won or lost.)
- **Data Preparation:** Once you have structured your data correctly, you should be following these preprocessing guidelines:
  - **Missing Data:** Intelligently handling missing data can improve model performance. Rows missing values in certain columns should be dropped or filled in an intelligent way (e.g., mean or median of column.)
  - **Check and remove multicollinearity:** You should be very careful about having redundancy in your data. For example, suppose you have two predictor columns “Sales” and “Revenue” and you want to predict “Profit.” Because “Sales” and “Revenue” represent nearly identical quantities (cash flow coming into the business), you should only include one of them. Using both can cause numerical stability issues in some algorithms. It is very common for DataSets to have

columns that are unexpectedly highly correlated, so it can be useful to visualize the relationships between the columns or test for correlations statistically. One of the correlated columns should be dropped.

  - **Include the relevant columns only:** You should drop any columns that you do not suspect to influence your output variable. ID and raw date columns likely fall into this category. Think carefully about what you know about the process and if your data correctly represents what you would know at prediction time.
  - **Outliers / Leverage Points:** If possible, you should remove known outliers. You can determine outliers either from statistical tests or known experience. For example, if sales were artificially high one week due to some known anomaly, you may want to consider removing this sample from your DataSet. For more information on handling and identifying outliers, see here.
- **Feature Engineering:** Finally, you should spend time thinking about the best representation of your data. This process is referred to as "feature engineering." Fortunately, AutoML does some basic feature engineering for us. For example, it will encode a "gender" column with values "M" and "F" as 0's as 1's. This allows us to focus more on the prior knowledge you have about your problem. A common example of the feature engineering you should think about is transforming raw data such as 10-22-2020 to the day of the week: "Thursday." This is a more helpful representation as it enables the model to learn patterns that might correspond to the day of the week. Other common feature engineering strategies are enumerated below:
  - **Binning:** If meaningful categories exist from numeric data, you may consider creating discrete "bins." A good example of this is the credit scoring system. While a FICO credit score offers a continuous, numeric value (0-800), banks will often create bins or tiers to facilitate easier decision making. A score over 750 might be considered "excellent," while a score between 700 and 749 is "good." If natural tiers exist in your problem, it may be helpful to encode them in your data. Additionally, time-series data can also be manipulated by binning. Dates across multiple years can be binned into the week of the year.
  - **Data Reductions:** A good model will have the fewest number of features that explain the data. Wherever possible, you should seek to remove redundant or unimportant columns. By including meaningless features, you add noise rather than signal to your model.

Following this checklist will improve your chances of training an effective model with AutoML.

Using AutoML

To use AutoML, you need AutoML enabled for your instance. See the headings Required Grants and Enable AutoML for instructions.

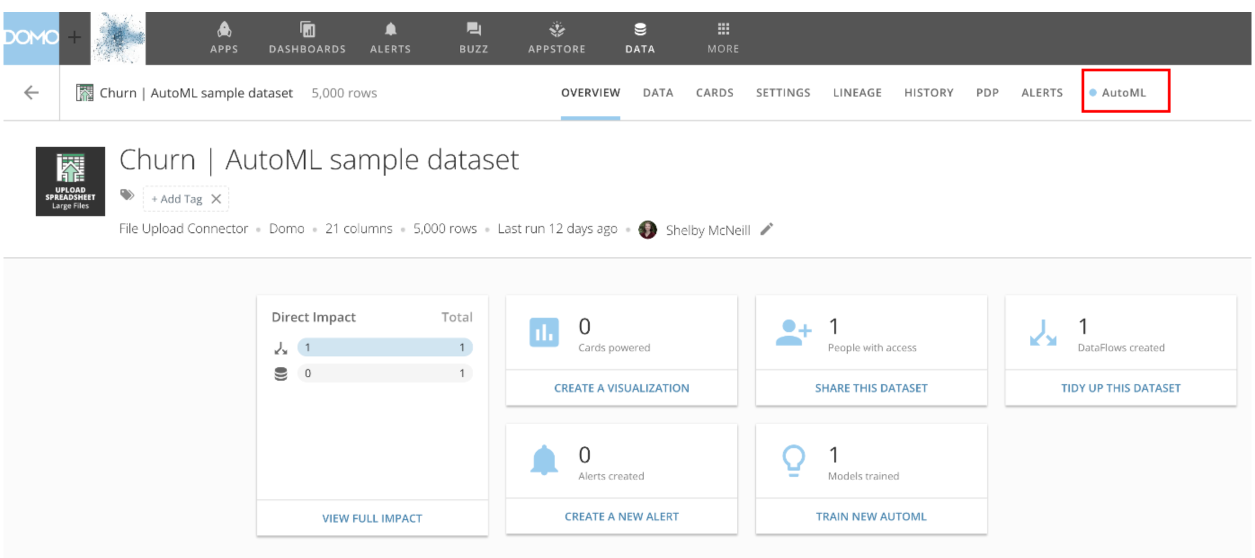
There are two parts to using AutoML: first, you launch an AutoML training job, which will generate multiple machine learning models specific to your data that you can then choose from; and second, you deploy your chosen model on a Domo DataSet to generate predictions using the AutoML Inference tile in a Magic ETL DataFlow.

A sample DataSet for you to try out AutoML can be downloaded at the following link. The sample DataSet includes data on customer churn at a phone company. Download the sample DataSet to your computer and then upload it to your Domo instance as a new DataSet.

Launch an AutoML Training Job

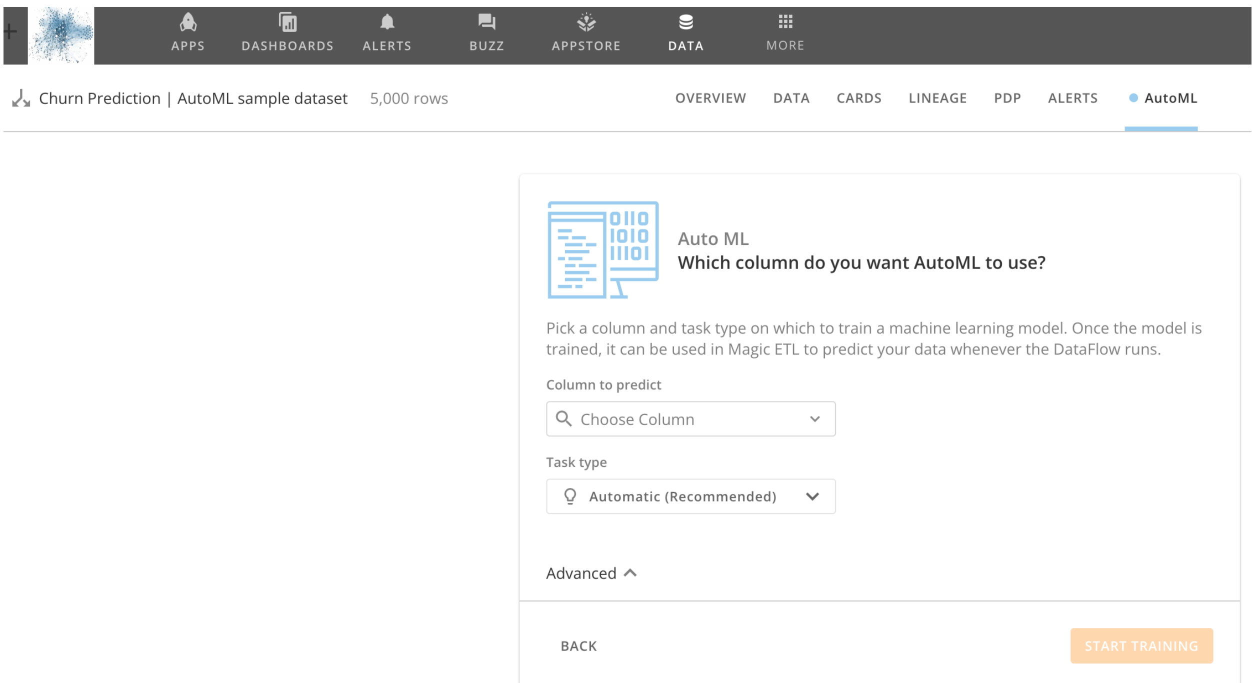
Below are instructions on how to launch an AutoML training job, which will generate multiple machine learning models specific to your data that you can then choose from.

1. Go to the DataSet Details page of the DataSet you would like AutoML to train machine learning models on. Select the **AutoML** tab.



2. Select the orange **Get Started** button.

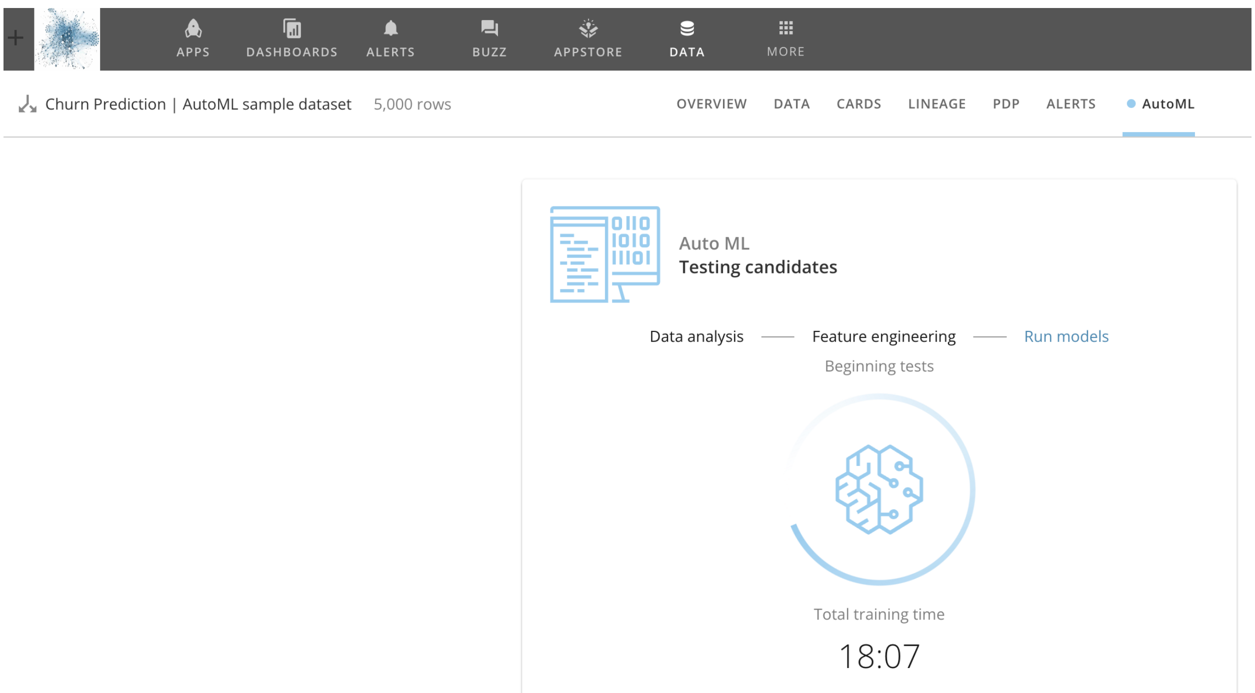
3. Using the **Column to predict** dropdown, select the output column on which the machine learning models will be trained to predict (if you're using the sample dataset, select the "class" column). Leave the **Task type** dropdown set to "Automatic (Recommended)" unless you're an advanced user who knows which task to specify. When you're ready to proceed, select the orange **Start Training** button.



4. A screen with a live “Total training time” counter should appear while AutoML is training and testing machine learning models using your data. AutoML specifically runs through the following three stages: Data analysis, Feature engineering, and Run models. The stage AutoML is currently in will be highlighted in blue.

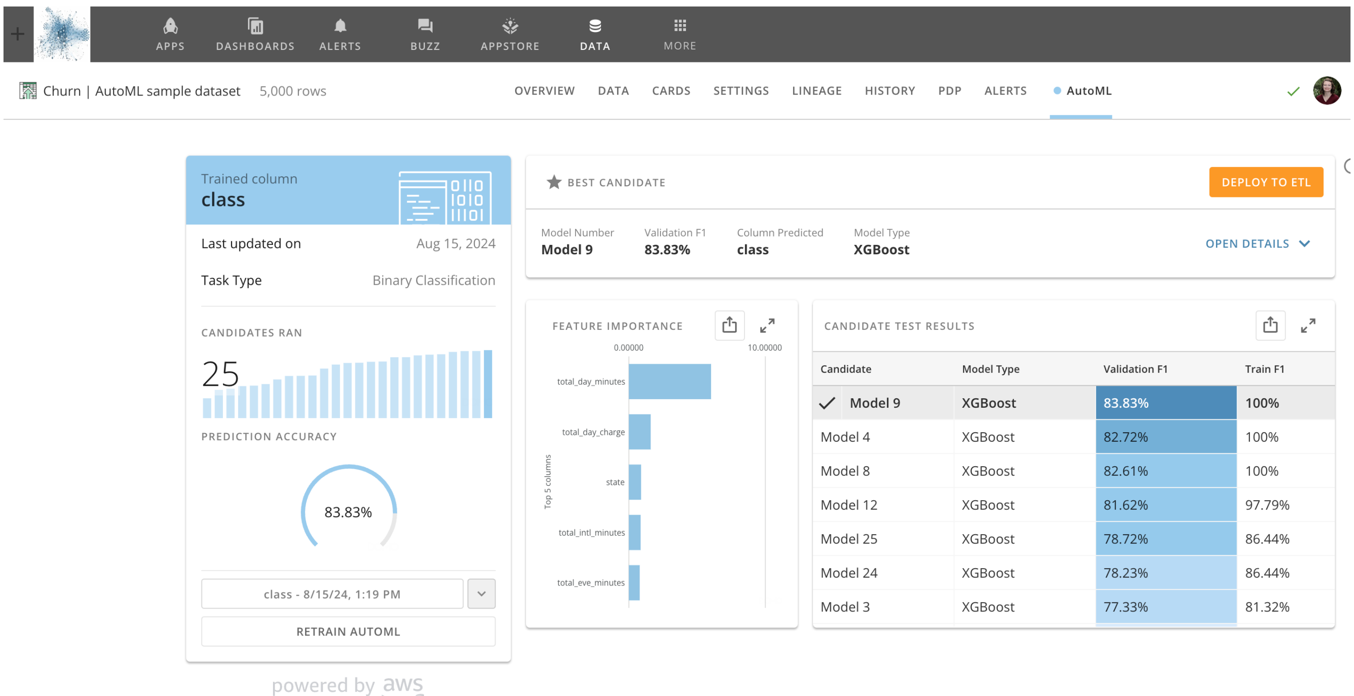
   Total training time can range from a few minutes to multiple hours depending on various factors, including the number of rows in your DataSet, the number of columns in your Dataset, the amount of feature engineering conducted by AutoML, task type, and the number of candidate models trained and tested. Generally, the larger the dataset or number of models trained, the longer the training time.

   If you’re trying out AutoML using the provided sample dataset and selected the “Automatic (Recommended)” task type, training time should be approximately 45 minutes.



5. Once AutoML is done training and testing machine learning models using your data, a Model Overview Page will appear. On this page, you can compare the performance of the models AutoML built. The top performing model will automatically be highlighted for you in the “Best Candidate” section.

   For additional background information on some of the metrics displayed in the Model Overview Page, see this Knowledge Base article on [Machine Learning Concepts to Help You be More Successful]().

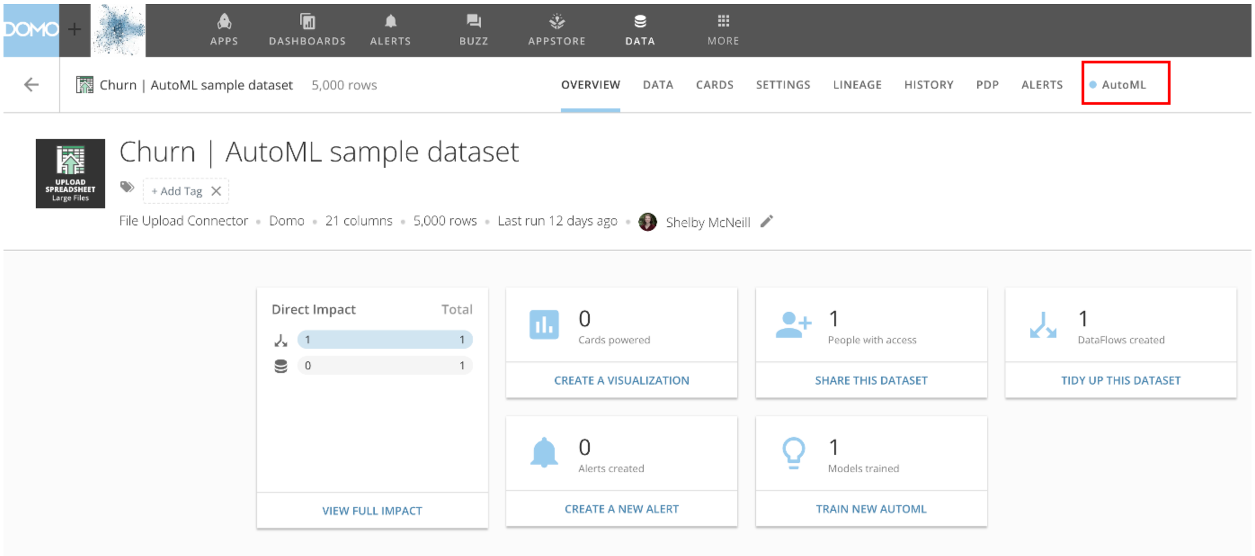


Generate Predictions Using the AutoML Inference Tile in Magic ETL

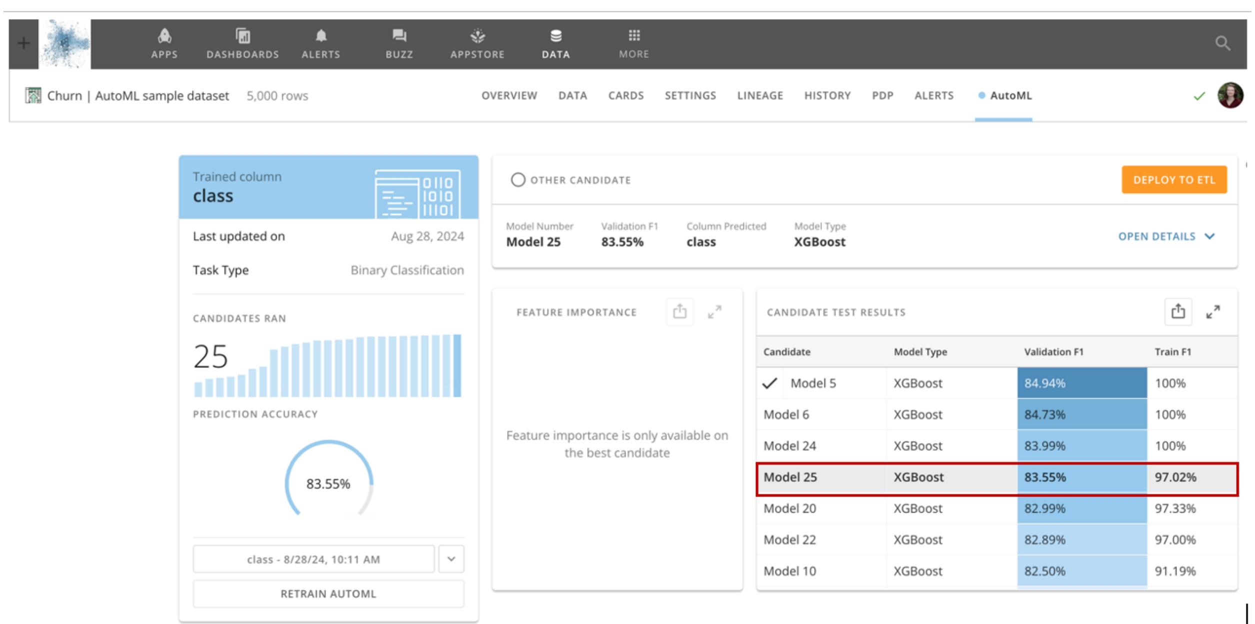
Prior to generating prediction usings the AutoML Inference tile, you need to have previously launched an AutoML training job. This will generate multiple machine learning models specific to your training data that you can then choose to deploy using the AutoML Inference tile.

Below are instructions on how to deploy your chosen AutoML model on a Domo DataSet using the AutoML Inference tile in a Magic ETL DataFlow. The AutoML Inference tile allows you to select and use a previously-trained AutoML machine learning model to make predictions (inference) for each row of your Input DataSet.

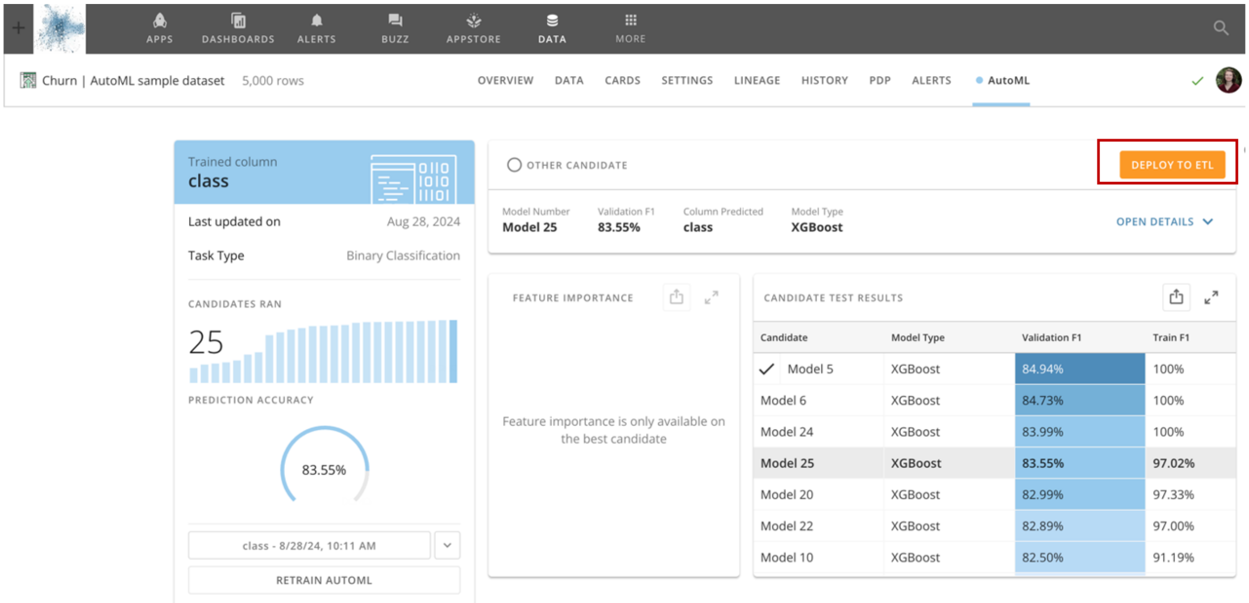
1. Go to the DataSet Details page of the DataSet you had AutoML train machine learning models on. Select the **AutoML** tab.



2. On the Model Overview Page, select the specific model under the “Candidate Test Results” section that you would like to use to generate predictions via the AutoML Inference tile. Your selected model will be highlighted in gray. In the example below, Model 25 has been selected.



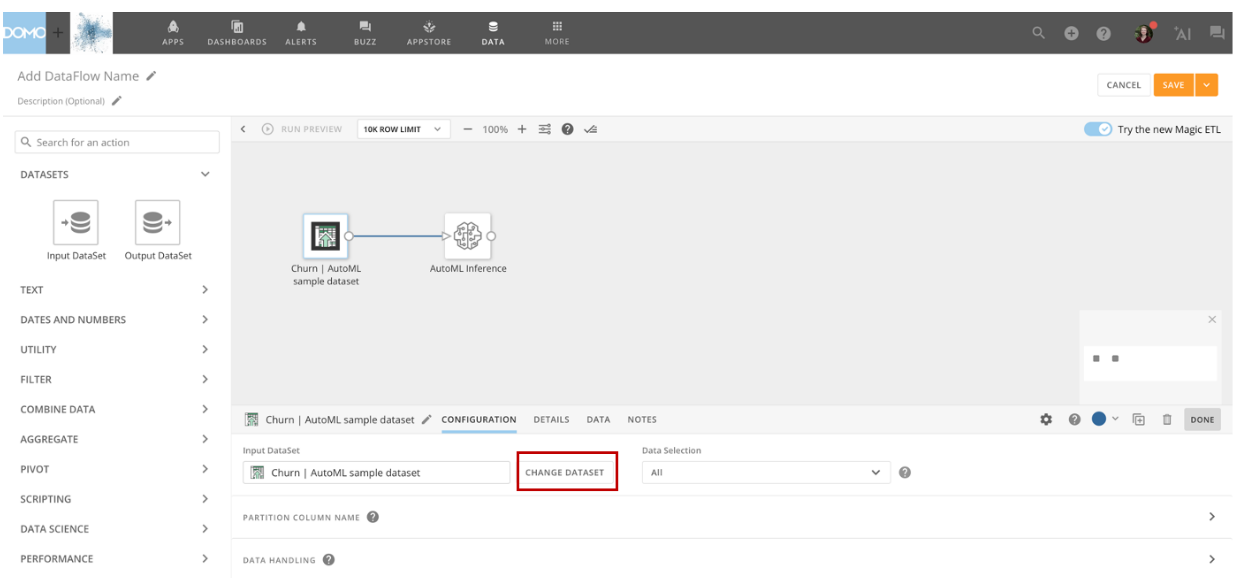
3. Select the **Deploy to ETL** orange button in the top right-hand corner of the Model Overview Page to deploy your model and be redirected to the Magic ETL interface.



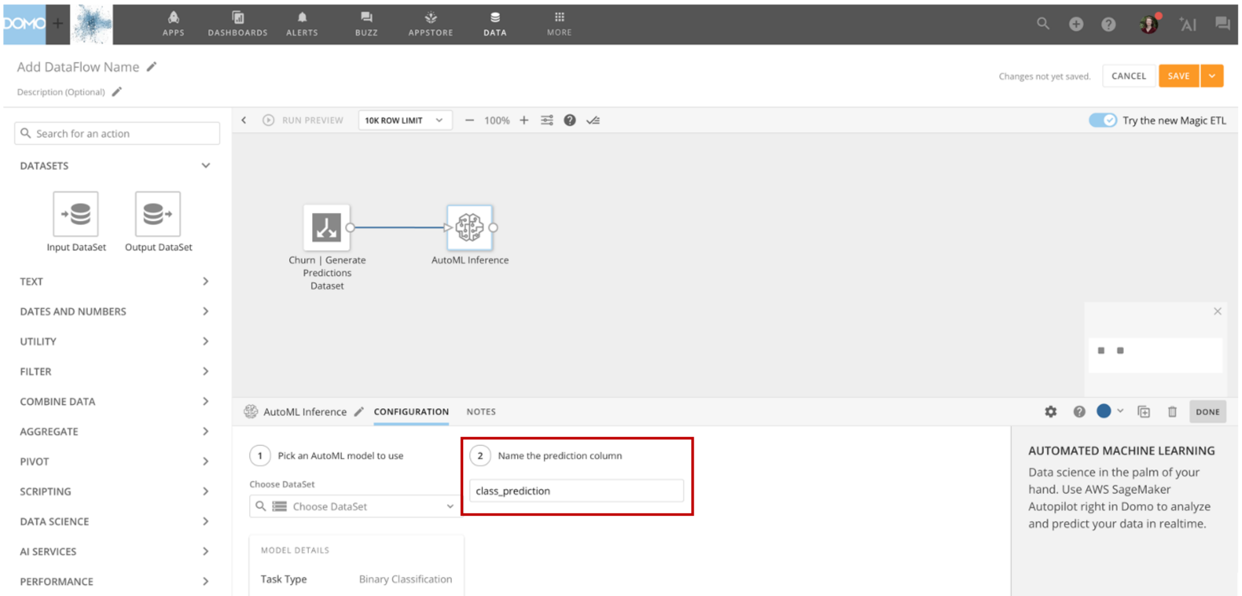
You can also deploy your chosen AutoML model by opening a new Magic ETL DataFlow, dragging an AutoML Inference tile onto the canvas from the Data Science section of the left rail, and then configuring the AutoML Inference tile. However, when you use this method of model deployment, you can only deploy the model that is deemed best by AutoML (this "best" model is the model listed first under the "Candidate Test Results" section of the Model Overview page).

4. Once you are redirected to the Magic ETL interface, you should see an Input DataSet tile and the AutoML Interface tile pre-populated in a DataFlow. Change the Input DataSet tile from connecting to the DataSet you used to train the model to the DataSet you would like to use to generate predictions by selecting the **Input DataSet** tile and then selecting **Change DataSet** from the tile configurations.

   The Input DataSet that you would like to use to generate predictions must have the same schema— columns with the same names and data types—as the dataset used to train the AutoML model. For example, if your training DataSet had columns named *education_level*, *hourly_pay_rate*, and *tenure* that were used to predict your Outcome column, then the Input DataSet you select should also have these three columns. Further, if in your training DataSet *education_level* is a Text variable and *hourly_pay_rate* and *tenure* are Integer variables, then *education_level* should be a Text variable and *hourly_pay_rate* and *tenure* should be Integer variables in the Input Dataset you select. The Input DataSet does not have to contain the Outcome column that was predicted in the training DataSet. If the schema from the Input DataSet you select doesn't match the schema of the dataset used to train the AutoML model, an error message will appear and display the mismatching details.



5. (Optional) When you select the **AutoML Inference tile**, you'll see that the "Name the prediction column" option has been auto-populated for you. You can type in another name for this column if you desire.



6. Connect and name an **Output Dataset**. Lastly, name your DataFlow and **Save and Run** the dataflow.
   The resulting output DataSet will include all columns from the Input DataSet plus a column (named in Step 5 above) that contains the predictions.