



Tutorial | Causal prediction

Although [interactive what-if analysis](#) can provide fast answers about a model's prediction given a new set of input values for exploration purposes, it does not provide any guardrails nor a way to evaluate the causal assumptions.

However, causal prediction (also known as uplift modeling) helps you quantify cause and effect relationships by modeling the differences between outcomes with and without controllable actions — or treatments — applied, so you can make better decisions and improve business results. In other words, it helps you determine the causal effect of an action or treatment on some outcome of interest.

This AutoML feature makes it simple to set up the prediction experiment, develop an optional treatment propensity model, and evaluate causal performance metrics, all in Dataiku's familiar visual ML framework.

Get started

Objectives

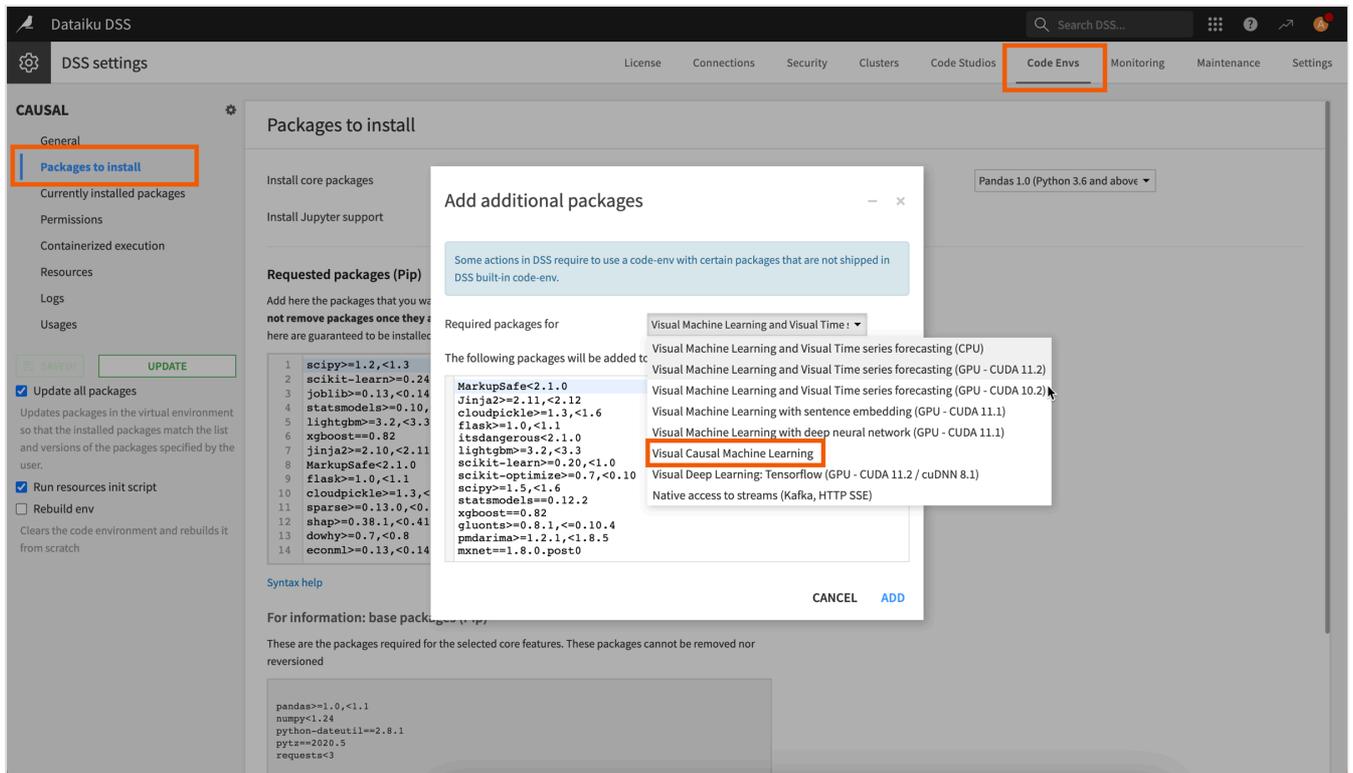
In this tutorial, you will:

- Create and configure a causal prediction model with one or multiple treatments.
- Train it.
- Evaluate it.
- Score new data with it.

Prerequisites & limitations

To complete this tutorial, you'll need:

- Dataiku 12.0 or later.
- A Full Designer user profile.
- A compatible [code environment](#) to train and run causal models. This environment must be created beforehand by an administrator and include the **Visual Causal Machine Learning** package.



Warning

Causal prediction is incompatible with the following:

- MLflow models, custom models & custom metrics
- Models ensembling
- Model export, Model Document Generator
- SQL, Spark or Java (optimized) scoring
- Model Evaluation Stores

Use case summary

Let's say our marketing team wants to tackle a customer churn problem.

We want to offer a renewal discount to only the customers most likely to respond positively to a promotional campaign since it's costly — and counterproductive — to distribute the offer to everyone.

How can we effectively prioritize which customers to treat with this promotion? Dataiku has a dedicated AutoML task for causal predictions, so we can measure the likely difference in

outcome with and without the marketing treatment.

Create the project

Dataiku 13.3+

Dataiku Pre-13.3

1. From the Dataiku Design homepage, click **+ New Project**.
2. Select **Learning projects**.
3. Search for and select **Causal Prediction**.
4. Click **Install**.
5. From the project homepage, click **Go to Flow** (or **g** + **f**).

Note

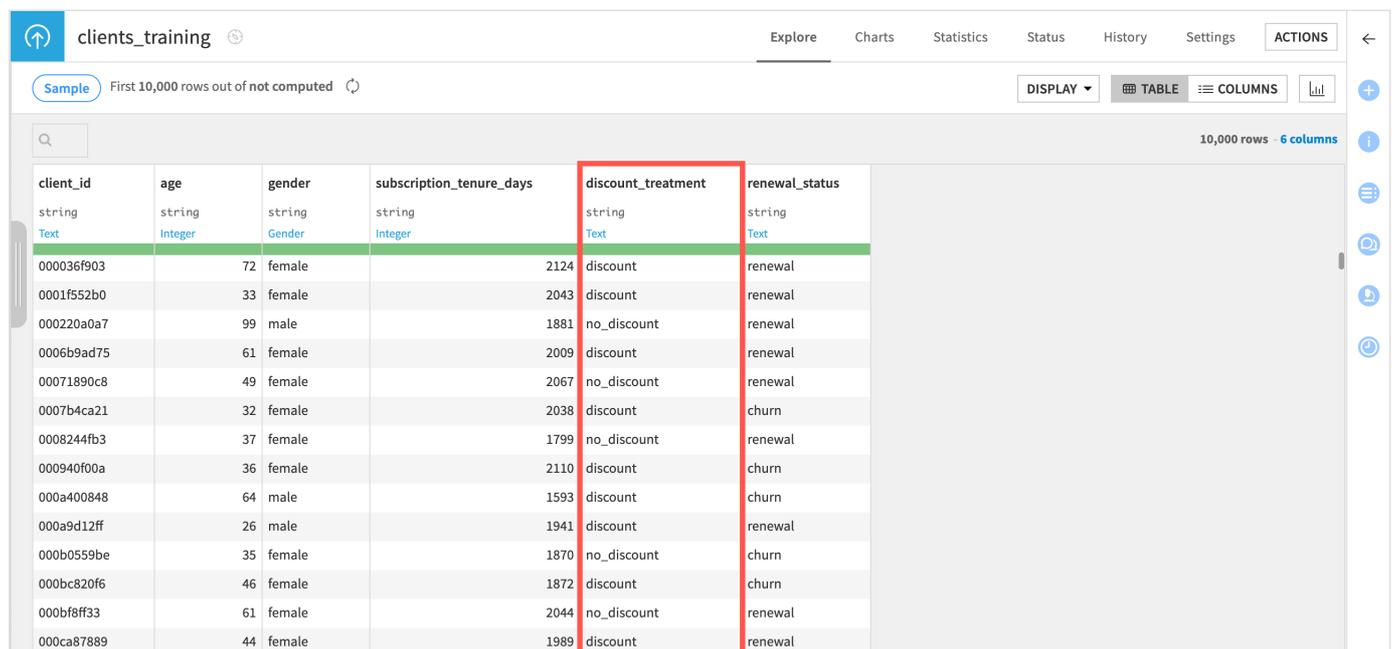
You can also download the starter project from this [website](#) and import it as a zip file.

Once you have this project, you can decide whether you want to use one or multiple treatments.

Single treatment

From the Flow:

1. Double-click on the **Causal prediction - single treatment** Flow zone to focus on this part of the project.
2. Open the **clients_training** dataset and look at the values in the *discount_treatment* column, which we'll use later on to indicate if the clients received a discount or not.



The screenshot shows the Dataiku interface for the 'clients_training' dataset. The table has 6 columns: client_id, age, gender, subscription_tenure_days, discount_treatment, and renewal_status. The 'discount_treatment' column is highlighted with a red box. The table displays the first 10,000 rows out of not computed.

client_id	age	gender	subscription_tenure_days	discount_treatment	renewal_status
000036f903	72	female	2124	discount	renewal
0001f552b0	33	female	2043	discount	renewal
000220a0a7	99	male	1881	no_discount	renewal
0006b9ad75	61	female	2009	discount	renewal
00071890c8	49	female	2067	no_discount	renewal
0007b4ca21	32	female	2038	discount	churn
0008244fb3	37	female	1799	no_discount	renewal
000940f00a	36	female	2110	discount	churn
000a400848	64	male	1593	discount	churn
000a9d12ff	26	male	1941	discount	renewal
000b0559be	35	female	1870	no_discount	churn
000bc820f6	46	female	1872	discount	churn
000bf8ff33	61	female	2044	no_discount	renewal
000ca87889	44	female	1989	discount	renewal

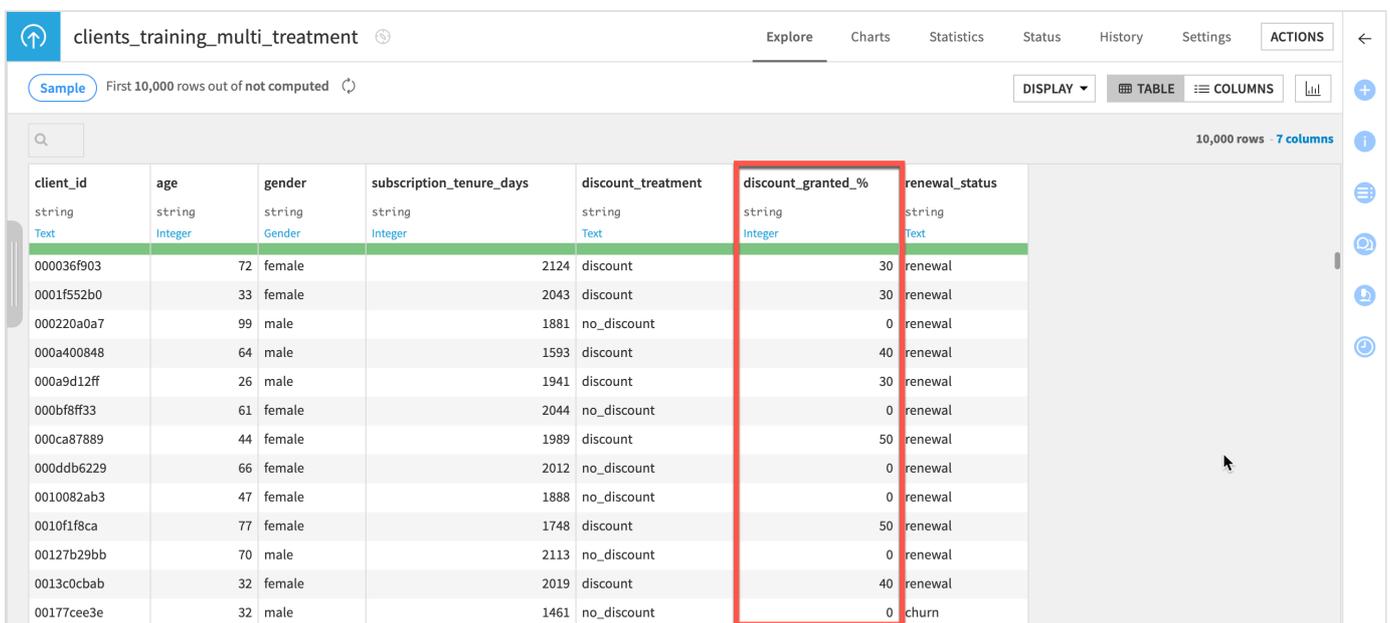
As you can see above, since this Flow is used to build a causal prediction model with a single treatment:

- There's only one value to indicate that a client received a discount: *discount*.
- The other value, *no_discount*, is our control value.

Multiple treatments

From the Flow:

1. Double-click on the **Causal prediction - multiple treatment** Flow zone to focus on this part of the project.
2. Open the **clients_training_multi_treatment** dataset and look at the values in the *discount_granted_%* column, which we'll use later on to indicate if the clients received a discount or not, and the amount of said discount.



The screenshot shows a data table with the following columns and data:

client_id	age	gender	subscription_tenure_days	discount_treatment	discount_granted_%	renewal_status
000036f903	72	female	2124	discount	30	renewal
0001f552b0	33	female	2043	discount	30	renewal
000220a0a7	99	male	1881	no_discount	0	renewal
000a400848	64	male	1593	discount	40	renewal
000a9d12ff	26	male	1941	discount	30	renewal
000bf8ff33	61	female	2044	no_discount	0	renewal
000ca87889	44	female	1989	discount	50	renewal
000db6229	66	female	2012	no_discount	0	renewal
0010082ab3	47	female	1888	no_discount	0	renewal
0010f1f8ca	77	female	1748	discount	50	renewal
00127b29bb	70	male	2113	no_discount	0	renewal
0013c0cbab	32	female	2019	discount	40	renewal
00177cee3e	32	male	1461	no_discount	0	churn

As you can see above, since this Flow is used to build a causal prediction model with multiple treatments:

- The column includes several amounts for the discount received by clients who received the treatment: *30%*, *40%* or *50%*.
- For clients who were not treated, *0* is our control value.

Create the causal prediction

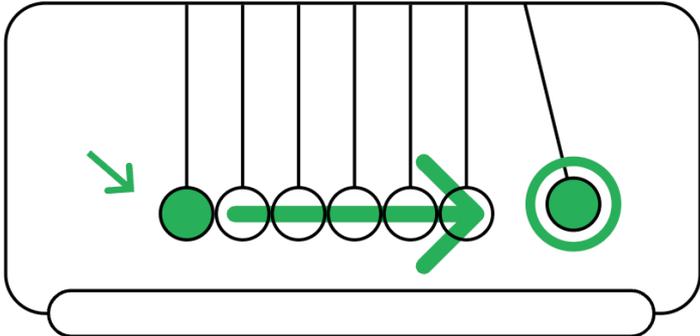
Let's create the causal prediction model.

Single treatment

To create a causal prediction model with only one treatment, follow the steps below:

1. From the **Causal prediction** Flow zone, select the **clients_training** dataset, and in the **Actions** panel on the right, click **Lab > Causal Prediction**.
2. In the new window, select **discount_treatment** as the treatment and **renewal_status** as the outcome.
3. Click **Create**.

Create prediction of the effect of `discount_treat...` on `renewal_status`



Model the effect of a treatment on an outcome

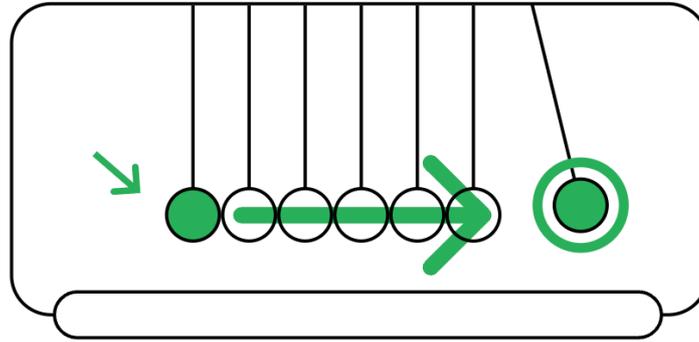
Name your analysis [CREATE](#)

Multiple treatments

To create a causal prediction model with multiple treatments, follow the steps below:

1. From the **Causal prediction** Flow zone, select the **clients_training_multi_treatment** dataset, and in the **Actions** panel on the right, click **Lab > Causal Prediction**.
2. In the new window, select **discount_granted_%** as the treatment and **renewal_status** as the outcome.
3. Click **Create**.

Create prediction of the effect of `discount_grant...` on `renewal_status`



Model the effect of a treatment on an outcome

Name your analysis

Predict effect of "discount_granted_%" on "renewal_status"

CREATE

Configure the causal prediction model

In the **Design** tab, let's now configure the causal prediction model.

Set the outcome & treatment

First, let's configure the basic settings required for causal prediction. By default, you should already be into the **Outcome & Treatment** panel of the **Basic** section.

Important

Selecting the appropriate outcome and treatment parameters is crucial for accurately calculating the predicted treatment effect, which is defined as the probability of **renewal with discount** minus the probability of **renewal without discount**.

If you misconfigure one of these parameters, you will end up with the opposite effect!

Single treatment

For this model using a single treatment, the **Outcome** option is already set to `renewal_status` and the **Treatment variable** to `discount_treatment` as we set them upon creating the causal prediction.

Configure the remaining options as follows:

1. In the **Outcome** section, set the **Preferred outcome class** to `renewal`, which means that the customer renewed their subscription.
2. In the **Treatment** section, set the **Control value** to `no_discount` which means that no offer is sent to the control population.
3. Click **Save**.

Outcome & Treatment

Outcome

Prediction type: Causal Classification

Outcome: renewal_status

Preferred outcome class: renewal

Predicted effects will be computed with the predicted probabilities of this class

Treatment

Treatment variable: discount_treatment

Control value: no_discount

Treatment is considered binary; either equal to the control value (control group) or not (treated group)

Allow multi-valued treatment:

Drop missing values:

Estimated Average Treatment Effect

Assuming full randomization of the treatment variable (`discount_treatment`), the **Average Treatment Effect (ATE)** can be estimated by the difference of the frequency of the preferred class `renewal_status` with treatment `discount_treatment` is not `no_discount` and without. The estimated ATE is **0.0024895**.

The values below are computed on a sample using the current sampling settings. You can edit them from the [analysis script](#).

	renewal	churn	All	Preferred class frequency
Treatment				
Control <i>discount_treatment is no_discount</i>	3044	1949	4993	0.60965
Treatment <i>discount_treatment is not no_discount</i>	3065	1942	5007	0.61214
All	6109	3891	10000	0.61090

Multiple treatments

For this model using multiple treatments:

- The **Outcome** option is already set to **renewal_status** and the **Treatment variable** to **discount_granted_%** as we set them upon creating the causal prediction.
- The **Allow multi-valued treatment** option is checked.

Configure the remaining options as follows:

1. In the **Outcome** section, set the **Preferred outcome class** to **renewal**, which means that the customer renewed their subscription.
2. In the **Treatment** section, ensure that the **Control value** is set to **0** which means that no offer is sent to the control population.
3. Click **Save**.

Predict effect of "discount_granted_" on "renewal_status"

Predict effect of discount_granted_% on renew... (Causal classification) DESIGN RESULT SAVE TRAIN

BASIC

Outcome & Treatment RE-DETECT SETTINGS

Outcome

Prediction type: Causal Classification

Outcome: renewal_status

Preferred outcome class: renewal

Predicted effects will be computed with the predicted probabilities of this class

Treatment

Treatment variable: discount_granted_%

Control value: 0

All values except the control value are considered as a different treatment

Allow multi-valued treatment:

Drop missing values:

Estimated Average Treatment Effect

Assuming full randomization of the treatment variable (discount_granted_%), the **Average Treatment Effect (ATE)** for each treatment can be estimated by the difference of the frequency of the preferred class renewal_status with and without treatment.

The values below are computed on a sample using the current sampling settings. You can edit them from the [analysis script](#).

	Outcome value	renewal	churn	All	ATE Preferred class frequency
Treatment					
Control discount_granted_% is 0		2772	1655	4427	- 0.62616
Treatment: 50 discount_granted_% is 50		1309	340	1649	0.16766 0.79381
Treatment: 40 discount_granted_% is 40		1060	603	1663	0.011245 0.63740
Treatment: 30 discount_granted_% is 30		935	578	1513	-0.0081801 0.61798
All		6076	3176	9252	0.058609 0.65672

BASIC

Outcome & Treatment

Train / Test Set

Metrics

Debugging

FEATURES

Features handling

Feature generation

Feature reduction

MODELING

Treatment analysis

Algorithms

Hyperparameters

ADVANCED

Runtime environment

Configure the other model settings

Important

From now on, whatever the project you're using in this tutorial (single or multiple treatments), the configuration remains the same.

For the sake of clarity, we're focusing on the project with a single treatment.

Set the train/test set

This section allows you to define the split policy upon training the data. By default, when you train the model:

- 80% of the data is used for training.
- 20% of the data is used for testing.

For this tutorial, we'll keep this default setting. However, we'll use the whole dataset and not just a sample. To do so:

1. Under the **Basic** section, select the **Train/Test Set** panel.
2. In the **Sampling & Splitting** section, set the **Sampling method** to **No sampling (whole data)**.
3. Click **Save**.

Configure the treatment analysis

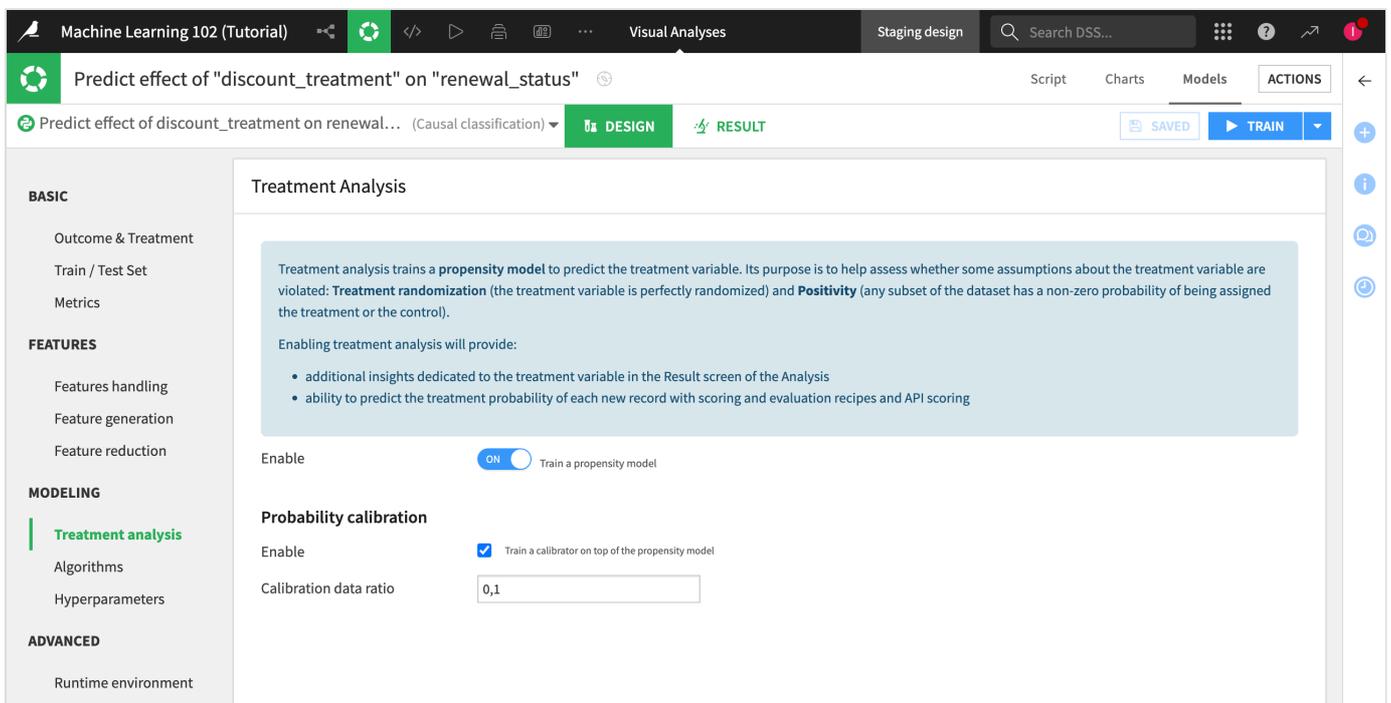
Though the design settings for causal prediction are similar to that of other prediction tasks, there are some notable differences, including the possibility of simultaneously training a treatment propensity model to predict the probability of receiving the treatment.

In our tutorial, we'll enable a treatment propensity model to predict each customer's likelihood of being treated with a discount offer.

We'll also enable probability calibration to predict treatment probabilities that are as well-calibrated as possible. This is required to test the positivity assumption because it helps detect if there are major differences between the treated vs. untreated customers in our training sample, which can lead to an unreliable model.

To do so:

1. Under the **Modeling** section, select the **Treatment Analysis** panel.
2. Turn the propensity modeling on.
3. Enable the use of a calibration model.
4. Click **Save**.



The screenshot shows the DSS interface for a causal classification task. The main panel is titled "Treatment Analysis" and contains the following information:

- Enable:** A toggle switch is set to "ON" with the label "Train a propensity model".
- Probability calibration:** A checkbox is checked with the label "Train a calibrator on top of the propensity model".
- Calibration data ratio:** A text input field contains the value "0,1".

A light blue informational box at the top of the panel explains the purpose of treatment analysis and lists the benefits of enabling it:

- additional insights dedicated to the treatment variable in the Result screen of the Analysis
- ability to predict the treatment probability of each new record with scoring and evaluation recipes and API scoring

The left sidebar shows the navigation menu with sections: BASIC, FEATURES, MODELING (where "Treatment analysis" is selected), and ADVANCED.

Select the algorithms

For [algorithms](#), you have a variety of causal methods to choose from, including:

- Meta learners, with a variety of base learners.
- Causal forest.

To configure the algorithms:

1. Under the **Modeling** section, select the **Algorithms** panel.
2. Let's keep the default settings (**T-learner** meta learner and **Causal forest**).

Machine Learning 102 (Tutorial) Visual Analyses Staging design Search DSS...

Predict effect of "discount_treatment" on "renewal_status" (Causal classification) DESIGN RESULT

BASIC

- Outcome & Treatment
- Train / Test Set
- Metrics

FEATURES

- Features handling
- Feature generation
- Feature reduction

MODELING

- Treatment analysis
- Algorithms**
- Hyperparameters

ADVANCED

- Runtime environment

Algorithms COPY TO... COPY FROM...

Meta-learners 2/13 algorithms
Meta-learners build models predicting the outcome variable, and use their predictions on the treated and control populations to infer the Conditional Average Treatment Effect. [Learn more](#)

S-learner T-learner X-learner

Causal Forest 1/1 algorithm
Causal forests build models predicting the treatment effect directly.

Base learners (2)

- Random Forest** ON
- Gradient tree boosting OFF
- Logistic Regression ON
- LightGBM OFF
- XGBoost OFF
- Decision Tree OFF
- Support Vector Machine OFF
- Stochastic Gradient Descent OFF
- KNN OFF

Random Forest ON

A **Random Forest** is made of many decision trees. Each tree in the forest predicts a record, and each tree "votes" for the final answer of the forest. [Show more...](#)

Number of trees 100
Number of trees in the forest.

Feature sampling strategy Default
Adjusts the number of features to sample at each split.

Maximum depth of tree 7
Maximum depth of each tree in the forest. Higher values generally increase the quality of the prediction, but can lead to overfitting. High values also increase the training and prediction time.

Minimum samples per leaf 25
Minimum number of samples required in a single tree node to split this node. Lower values increase the quality of the prediction (by splitting the tree node), but can lead to overfitting and increased training and prediction time.

Note

For more information on the different settings, see the [Causal Prediction Settings](#) page in the reference documentation.

Set the runtime environment

In the **Runtime environment** panel:

1. Indicate the correct [code environment](#) that includes the Visual Causal Machine Learning package.
2. Click **Save** to save all your changes.

Train the model

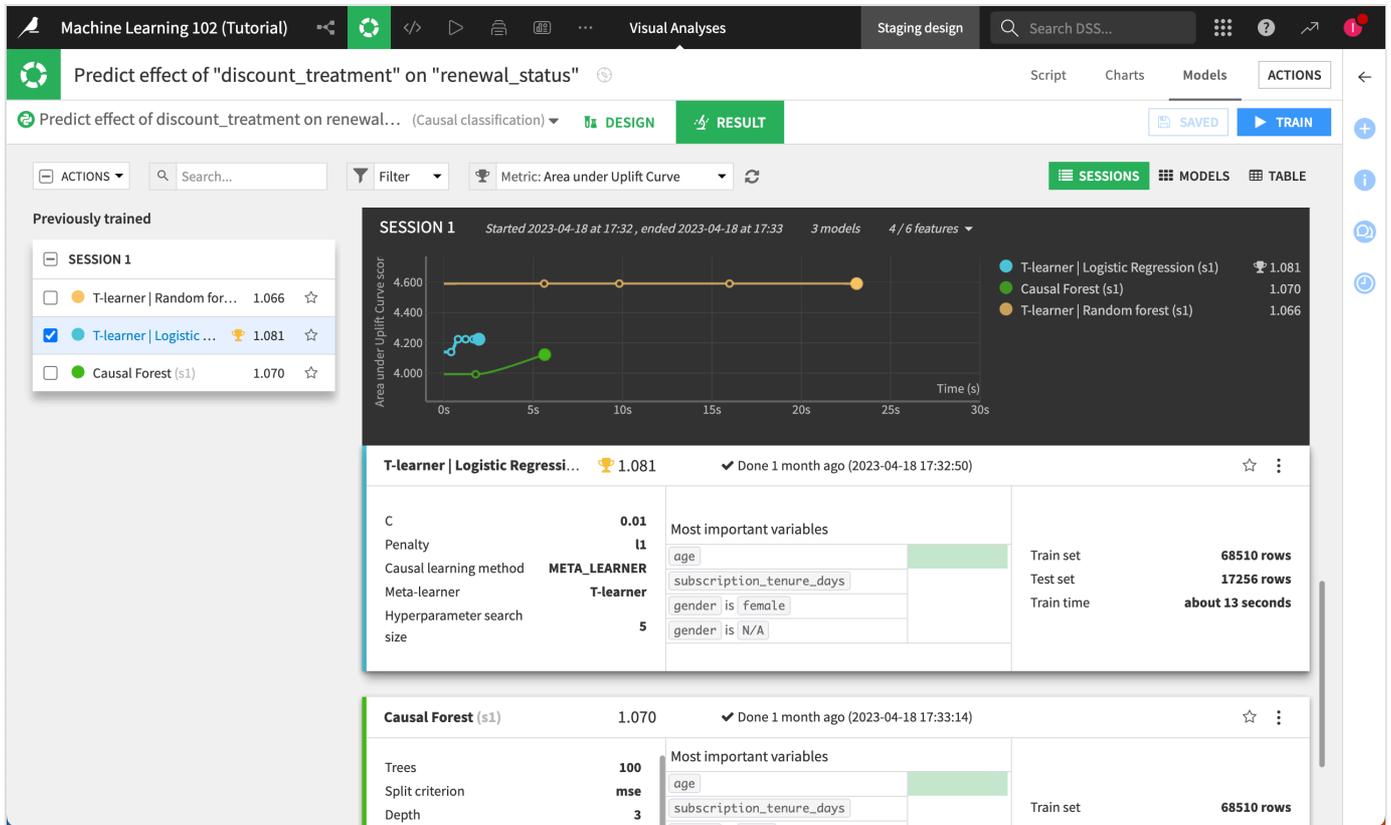
Once the model is configured, train it. To do so:

1. Click **Train** on top of the **Design** tab.
2. In the **Training models** dialog, optionally name and describe the model.
3. Click **Train** again.

Evaluate the training

As always, after we click train, let's head over to the **Results** tab to evaluate the models in our experiment.

Remember that unlike other types of predictive models you may have built with Dataiku, these evaluation metrics do not measure the model's ability to predict the outcome — in our case, renewal. Instead, they measure the model's ability to **predict the treatment effect on the outcome**. In other words, how well can this model predict the difference between subscription renewal with and without the promotional offer, all else being equal?



The score next to each model trained can be interpreted as below:

- A score lower than 0 indicates performance worse than random.
- A score around 0 (close to 0) suggests performance similar to random.
- A score higher than 0 indicates performance better than random.

The higher the score, the better the model's performance. Unlike other metrics like AUC, there is no upper bound for this metric, meaning higher values are preferred.

Since the **T-learner | Logistic Regression** model is the best-performing model here, let's click on it to check the results in greater detail.

Note

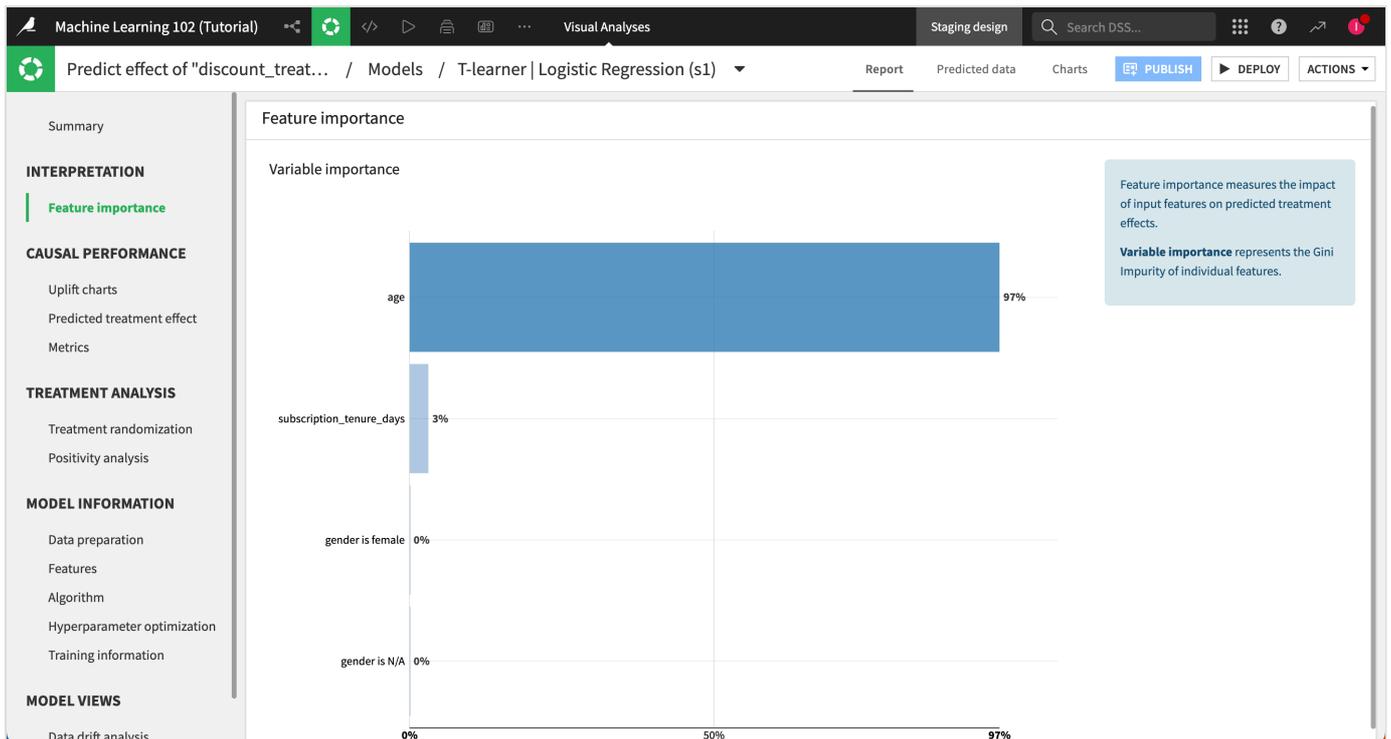
For causal prediction models with multiple treatments, training results are visible per each treatment value. In the model report, you must select a value to see the results in detail for this value.

Feature importance panel

Let's look at feature importance. The age of a subscriber being the important feature doesn't mean it has the strongest impact on the likelihood to churn, but rather that it has the strongest impact on how a customer reacts to promotions.

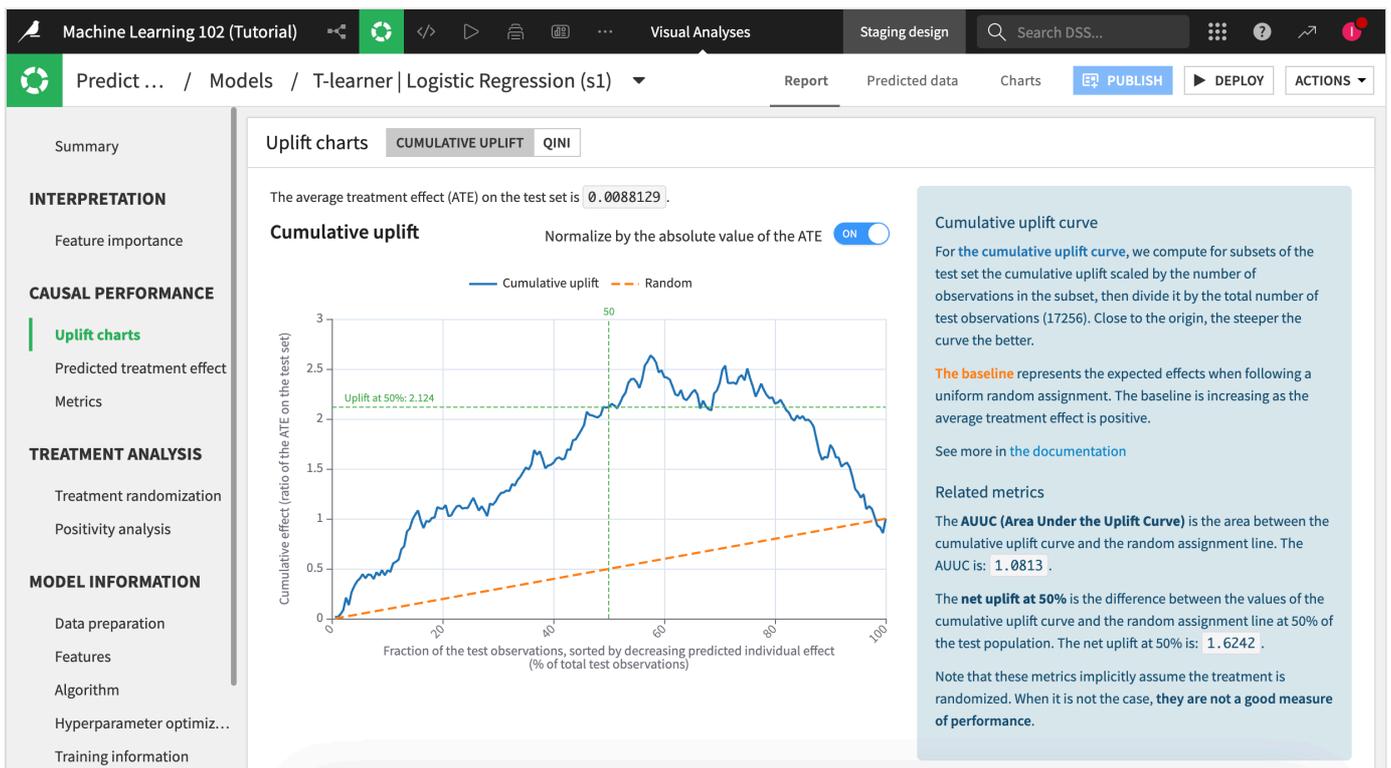
Note

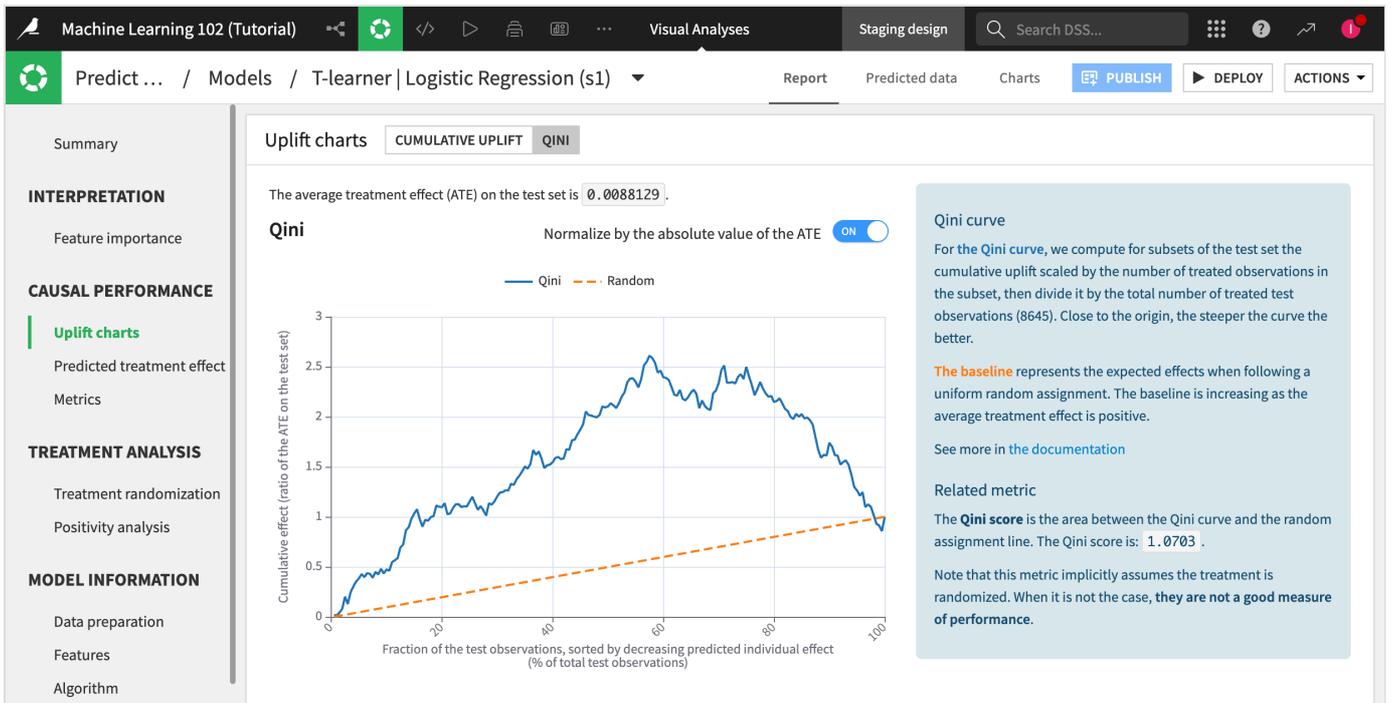
For more information, see the [Feature importance](#) section in the reference documentation.



Uplift charts panel

Uplift charts show us the conditional average treatment effect as compared to a random baseline. An upward curve shows a positive average effect of the treatment (i.e. the discount generally encourages people to renew their subscription) while a downward curve shows a negative average effect (i.e. offering the discount generally deters people from renewing their subscription).





Note

For more information, see the [Uplift and Qini curves](#) section in the reference documentation.

Predicted treatment effect panel

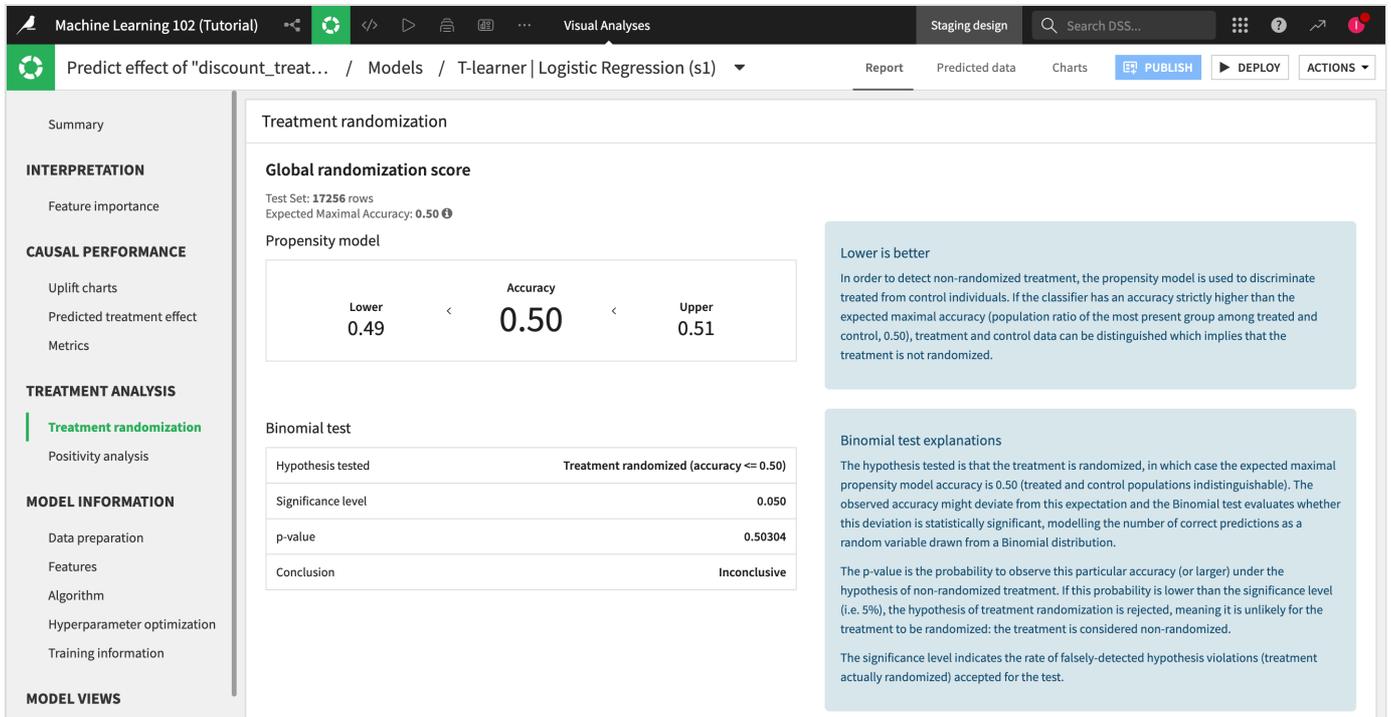
In the **Predicted treatment effect** panel, the histogram shows us the distribution of the predicted individual treatment effect. The X-axis shows the predicted effect while the Y-axis shows the count in each predicted-effect segment. We can use this chart to help us determine an appropriate predicted treatment effect cutoff threshold for our marketing action.

Notice that some customers have a negative treatment effect, so sending them the promotion would actually be counterproductive to our renewal goals.



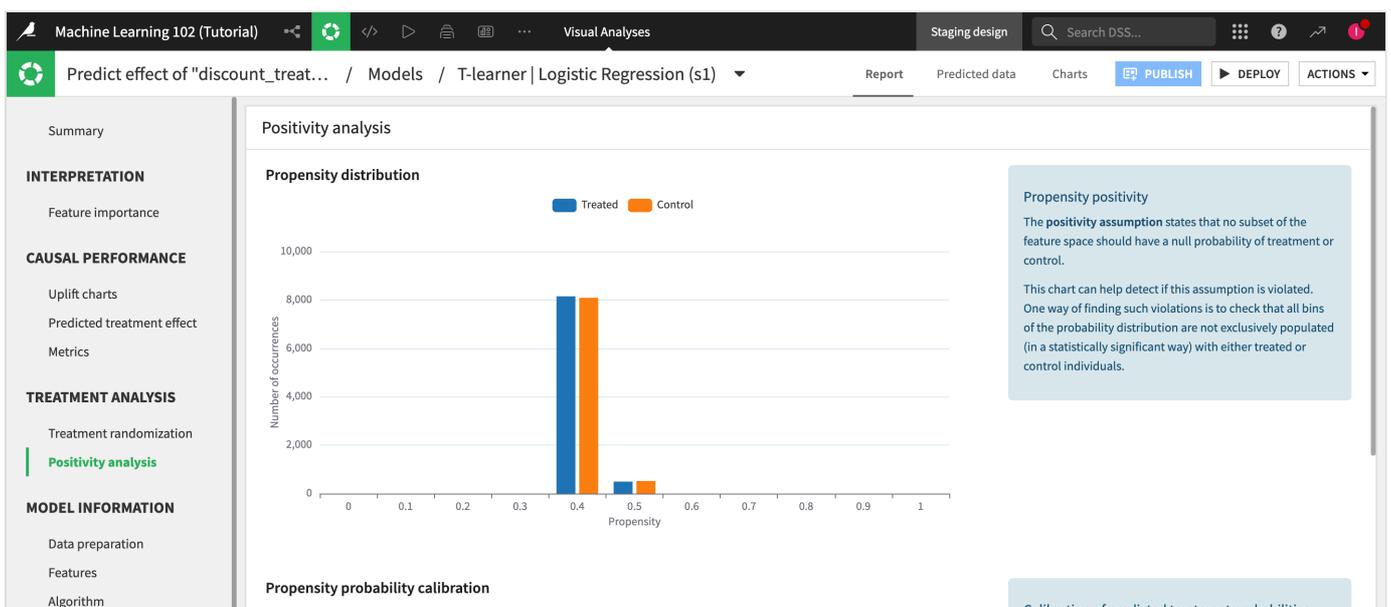
Treatment randomization panel

To achieve reliable results, remember to use a training dataset with a randomized treatment allocation to ensure that some customers won't be systematically excluded from the offer for one reason or another. When you enable treatment analysis to the causal prediction analysis, this panel helps you assess your data for treatment randomization, to see whether you need to make any adjustments to your training dataset.



Positivity analysis panel

Use **positivity analysis** charts to further examine treatment assumptions and consistency between predicted and observed frequencies.



Deploy the model to the Flow

When you have sufficiently explored building models, the next step is to deploy one from the Lab to the Flow. Here, we'll deploy the T-learner using the logistic regression algorithm as it is the one with the most positive treatment effect.

1. From the **Result** tab of the modeling task, click on the **T-learner | Logistic Regression (s1) model** to open its summary page.
2. Click the **Deploy** button in the upper right corner.
3. Keep the default settings and click **Create**. You are automatically redirected to the Flow.

Evaluate the model

1. In the **Causal prediction** Flow zone, select **Predict effect of dicount_treatment on renewal_status**, which is your deployed model.
2. In the right panel, under the **Evaluate model on already-known data** section, click on the **Evaluate** recipe. This opens the **Evaluate a model's performance** window.
3. Choose **clients_evaluation** as your input dataset.
4. In the **Outputs** section, set an output dataset named `model_scored` and a metrics dataset named `model_metrics`.

Evaluate a model's performance

Inputs	Outputs
<p>Input dataset</p> <p>clients_evaluation DATASET - View</p>	<p>Output dataset</p> <p>model_scored (Managed)</p> <p>CHANGE</p>
<p>Prediction model</p> <p>Predict effect of discount_treatment on renewal_s</p>	<p>Metrics</p> <p>model_metrics (Managed)</p> <p>CHANGE</p>

CANCEL CREATE RECIPE

5. Click **Create Recipe**.
6. Keep the default settings and click **Run**.

The Evaluate recipe generates two output datasets.

- *Model_scored* in which the recipe has appended the predicted effect and propensity measure (i.e. the probability to be treated).

client_id	age	gender	subscription_tenure_days	discount_treatment	renewal_status	predicted_effect	propensity
00047b3720	53	male	1573	discount	churn	0.001903482714407545	0.49976307060495934
0007667c60	46	male	1707	no_discount	renewal	0.010198436002300637	0.49976307060495934
00078c508d	48	female	2057	no_discount	renewal	-0.0010953631844473621	0.49976307060495934
000990be82	34	female	2031	no_discount	churn	0.021603674457101985	0.49976307060495934
000b9905d8	40	female	1964	discount	renewal	0.013703750092121325	0.49976307060495934
000bc94494	54	male	1823	no_discount	churn	-0.005442304797579567	0.49976307060495934
000efde438	53	male	2034	no_discount	renewal	-0.008440197811667827	0.5045045045045045
0015aa77ce	53	female	1935	discount	renewal	-0.006310224329262315	0.49976307060495934
00174b3561	44	female	2053	no_discount	renewal	0.00527875616789264	0.49976307060495934
00184e8b0a	65	female	2097	no_discount	churn	-0.028354723155786687	0.49976307060495934
001a2412c6	48	female	1937	discount	renewal	0.0015935493234506826	0.49976307060495934
001cef2991	34	female	1697	no_discount	renewal	0.030081010284156884	0.49976307060495934
0020f90a83	39	female	1645	no_discount	renewal	0.023191062145677677	0.49976307060495934
0024e5fa85	52	male	1774	discount	renewal	-0.0011157871477875947	0.49976307060495934
002deadd7f	43	female	1565	discount	renewal	0.01859221424491042	0.49976307060495934
0031c96fbc	44	female	2005	discount	churn	0.0063836036135747065	0.49976307060495934
0031da7c7b	70	female	2055	no_discount	renewal	-0.03532238095535278	0.49976307060495934
0046e67e68	42	male	1881	discount	renewal	0.012498589083581257	0.49976307060495934
0047aed44a	34	male	1583	discount	renewal	0.03302249655054168	0.49976307060495934
004f98c1b4	36	female	1900	no_discount	renewal	0.0016750131802310376	0.49976307060495934

- *Model_metrics* that logs the performance of the active version of the model against the input dataset.

date	auuc	qini	netUplift	propensityAuc	propensityLogLoss	propensityCalibrationLoss
2023-04-18T15:57:03.046298Z	2.2667501352980235	2.266062170765237	3.0004208870873446	0.50035808117776	0.6931831662194509	0.001096958353411123

Score new data using the model

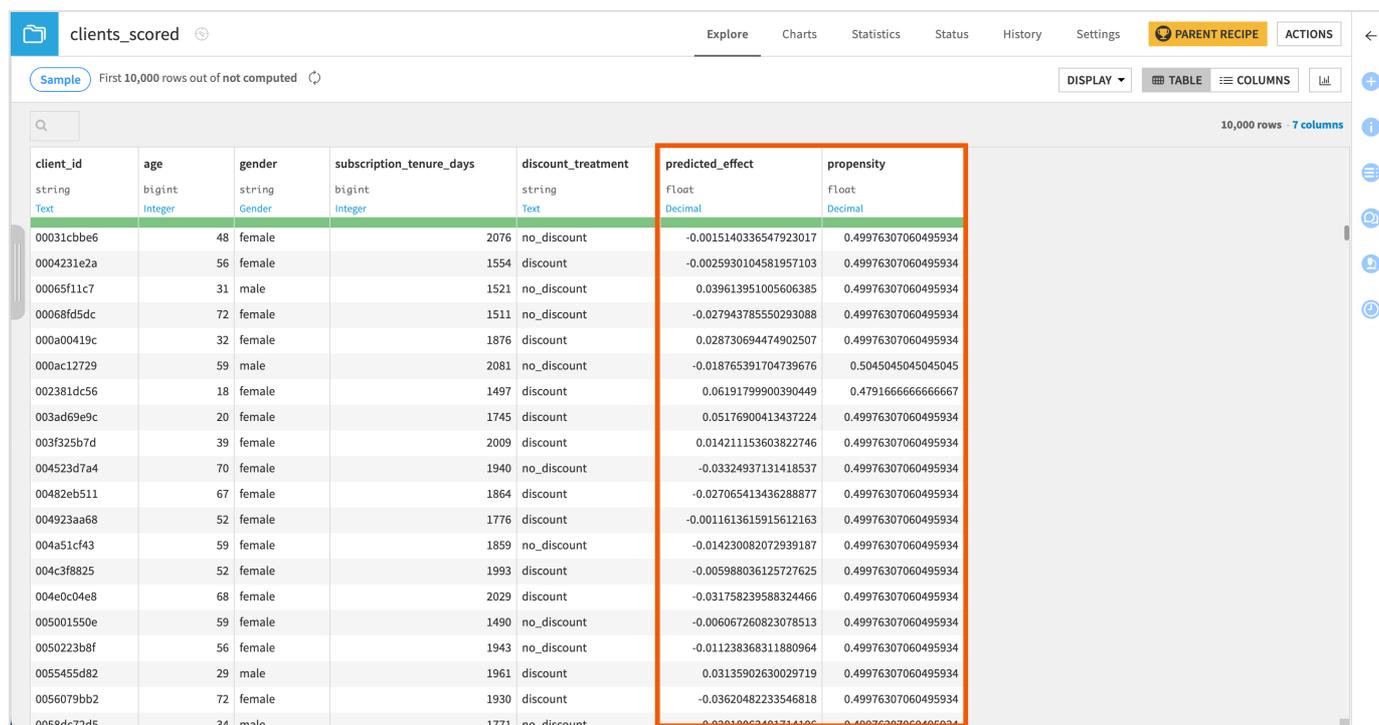
Run a Score recipe

Let's now use a Score recipe to apply our model to new and unseen data.

1. Go back to the Flow and click **Predict effect of discount_treatment on renewal_status** within the **Causal prediction** Flow zone.
2. In the right panel, under the **Apply model on data to predict** section, click on the **Score** recipe. This opens the **Score a dataset** window.
3. Choose **clients_scoring** as the input dataset.
4. Name the output dataset `clients_scored`.
5. Click **Create Recipe**.
6. Keep the default settings and click **Run**.

The Score recipe produces a dataset that appends two new columns to the input dataset:

- *predicted_effect* that predicts for each row whether the treatment has a positive, negative, or neutral impact on the customer.
- *propensity* that gives the probability for each customer to be treated.



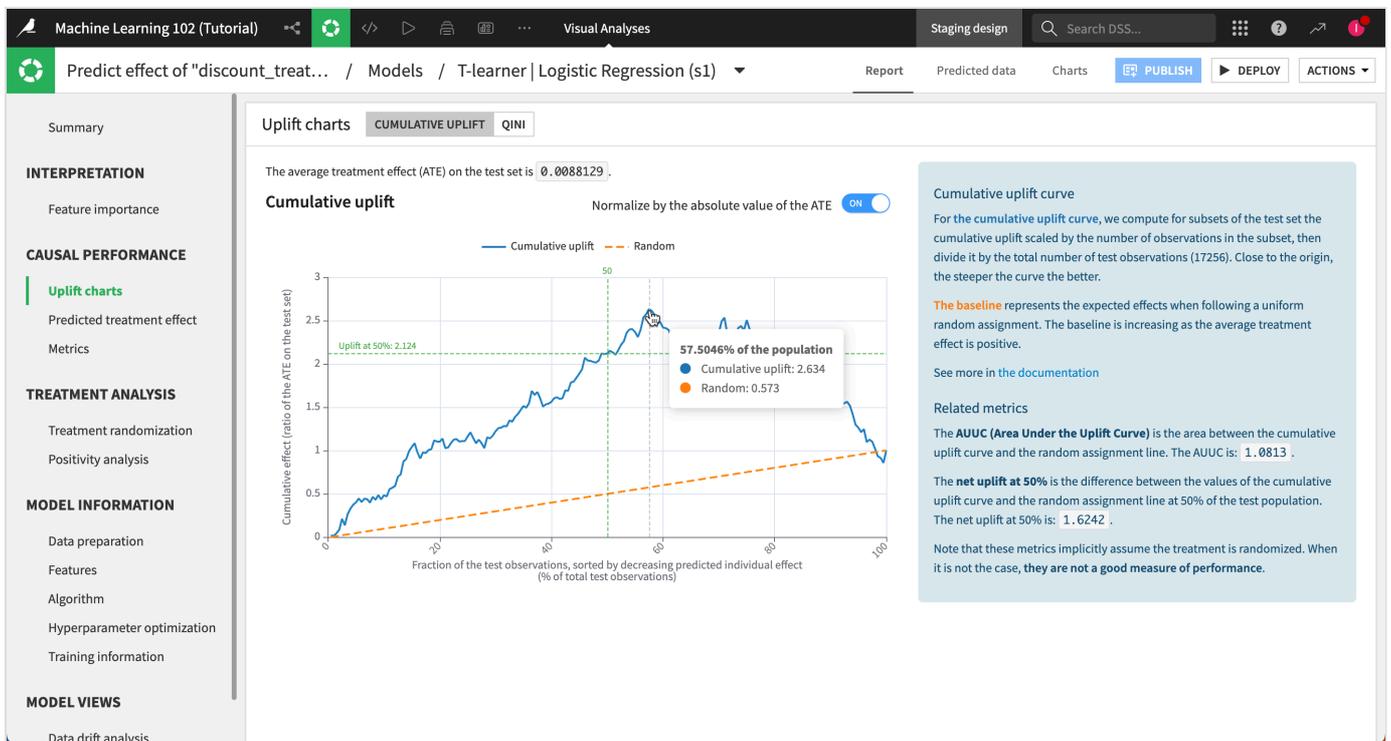
The screenshot shows a data table with 7 columns and 10,000 rows. The columns are: client_id (string), age (bigint), gender (string), subscription_tenure_days (bigint), discount_treatment (string), predicted_effect (float), and propensity (float). The predicted_effect and propensity columns are highlighted with a red box. The data is as follows:

client_id	age	gender	subscription_tenure_days	discount_treatment	predicted_effect	propensity
00031cbb6	48	female	2076	no_discount	-0.0015140336547923017	0.49976307060495934
0004231e2a	56	female	1554	discount	-0.0025930104581957103	0.49976307060495934
00065f1c7	31	male	1521	no_discount	0.039613951005606385	0.49976307060495934
00068fd5dc	72	female	1511	no_discount	-0.02794378550293088	0.49976307060495934
000a00419c	32	female	1876	discount	0.028730694474902507	0.49976307060495934
000ac12729	59	male	2081	no_discount	-0.018765391704739676	0.5045045045045045
002381dc56	18	female	1497	discount	0.06191799900390449	0.4791666666666667
003ad69e9c	20	female	1745	discount	0.05176900413437224	0.49976307060495934
003f325b7d	39	female	2009	discount	0.014211153603822746	0.49976307060495934
004523d7a4	70	female	1940	no_discount	-0.03324937131418537	0.49976307060495934
00482eb511	67	female	1864	discount	-0.027065413436288877	0.49976307060495934
004923aa68	52	female	1776	discount	-0.0011613615915612163	0.49976307060495934
004a51cf43	59	female	1859	no_discount	-0.014230082072939187	0.49976307060495934
004c3f8825	52	female	1993	discount	-0.005988036125727625	0.49976307060495934
004e0c04e8	68	female	2029	discount	-0.031758239588324466	0.49976307060495934
005001550e	59	female	1490	no_discount	-0.006067260823078513	0.49976307060495934
0050223b8f	56	female	1943	no_discount	-0.011238368311880964	0.49976307060495934
005455d82	29	male	1961	discount	0.03135902630029719	0.49976307060495934
0056079bb2	72	female	1930	discount	-0.03620482233546818	0.49976307060495934
0058dc72d5	34	male	1771	no_discount	-0.03010062401711106	0.49976307060495934

Specify a treatment ratio when scoring

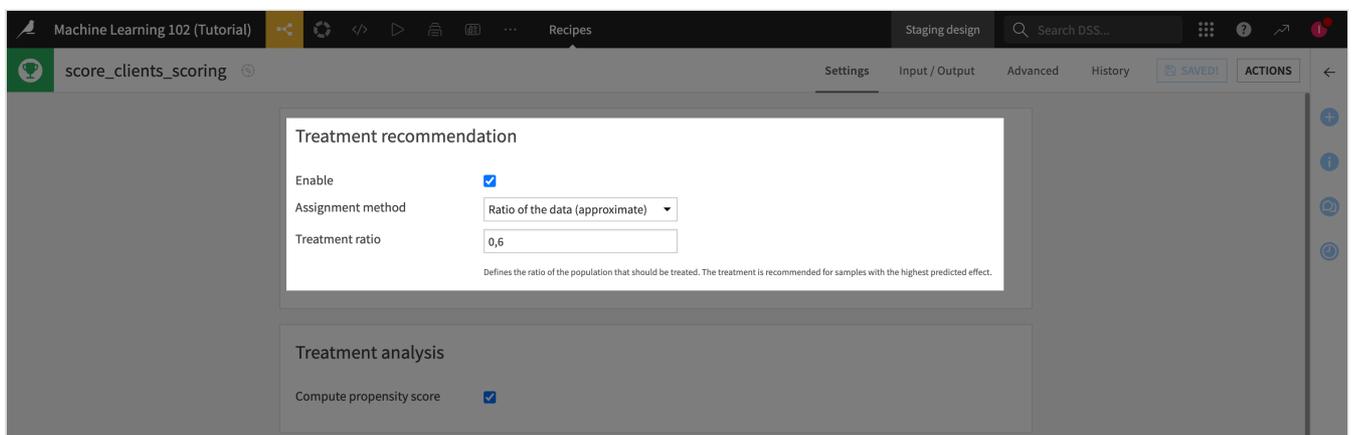
If you simply want to target percentage records with the highest predicted effect, you can specify a treatment ratio in the scoring recipe.

If you remember our uplift curve, the curve increased up to 57% of the population, which indicates that the treatment effect is positive on this part of the population. So in our tutorial, we'll score the dataset targeting this population.



To do so:

1. In the Flow, click on the Score recipe to open its settings page again.
2. Enable the treatment recommendation.
3. In the **Assignment method** option, select **Ratio of the data (approximate)**.
4. Set the **Treatment ratio** to $0,6$, namely the 60% of the population that interest us based on the uplift curve above.



5. Click **Run**.

Now, if you look at the *clients_scored* dataset, in addition to the *predicted_effect* and *propensity* columns, you have a third column named *treatment_recommended* that indicates for each customer whether the model recommends treating him/her with the discount.

clients_scored

Explore Charts Statistics Status History Settings PARENT RECIPE ACTIONS

Sample First 10,000 rows out of not computed

DISPLAY TABLE COLUMNS

10,000 rows 8 columns

client_id	age	gender	subscription_tenure_days	discount_treatment	predicted_effect	propensity	treatment_recommended
string	bigint	string	bigint	string	Float	Float	boolean
Text	Integer	Gender	Integer	Text	Decimal	Decimal	Boolean
00031cbb6	48	female	2076	no_discount	-0.0015140336547923017	0.49976307060495934	True
0004231e2a	56	female	1554	discount	-0.0025930104581957103	0.49976307060495934	False
00065f11c7	31	male	1521	no_discount	0.039613951005606385	0.49976307060495934	True
00068fd5dc	72	female	1511	no_discount	-0.027943785550293088	0.49976307060495934	False
000a00419c	32	female	1876	discount	0.028730694474902507	0.49976307060495934	True
000ac12729	59	male	2081	no_discount	-0.018765391704739676	0.5045045045045045	False
002381dc56	18	female	1497	discount	0.06191799900390449	0.4791666666666667	True
003ad69e9c	20	female	1745	discount	0.05176900413437224	0.49976307060495934	True
003f325b7d	39	female	2009	discount	0.014211153603822746	0.49976307060495934	True
004523d7a4	70	female	1940	no_discount	-0.03324937131418537	0.49976307060495934	False
00482eb511	67	female	1864	discount	-0.027065413436288877	0.49976307060495934	False
004923aa68	52	female	1776	discount	-0.0011613615915612163	0.49976307060495934	True
004a51cf43	59	female	1859	no_discount	-0.014230082072939187	0.49976307060495934	False
004c3f8825	52	female	1993	discount	-0.005988036125727625	0.49976307060495934	False
004e0c04e8	68	female	2029	discount	-0.031758239588324466	0.49976307060495934	False
005001550e	59	female	1490	no_discount	-0.006067260823078513	0.49976307060495934	False
0050223b8f	56	female	1943	no_discount	-0.011238368311880964	0.49976307060495934	False
0055455d82	29	male	1961	discount	0.03135902630029719	0.49976307060495934	True
0056079bb2	72	female	1930	discount	-0.03620482233546818	0.49976307060495934	False
0058dc72d5	34	male	1771	no_discount	0.02818063481714106	0.49976307060495934	True

What's next?

Congratulations! You're all set. You can now try it on your own use cases.