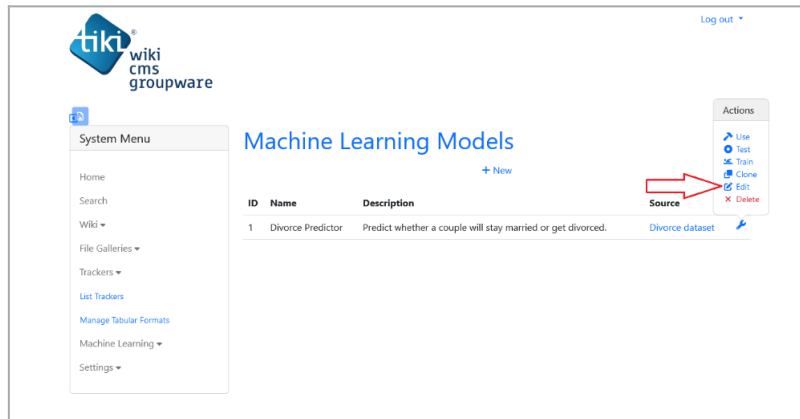


This page [needs review](#)

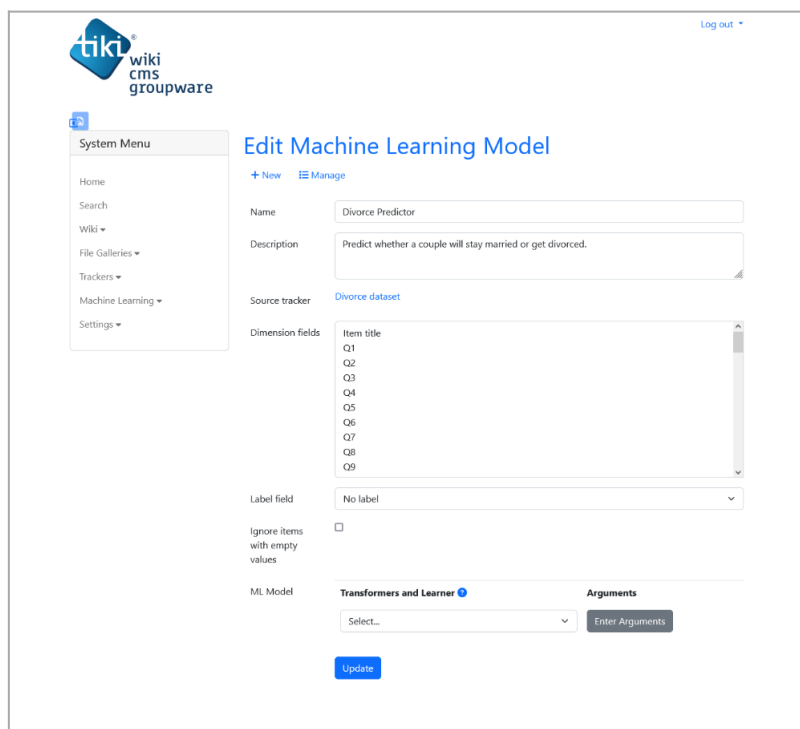
Machine Learning models are configured in accordance to the task they are to perform.

# Configuring Machine Learning Models

Configuring a model to make it ready for training involves specifying the data dimension fields, a label field if necessary, any required transformers and a learner. You can get to the model configuration page by finding the model in the Machine Learning **List Models** page, click on the model's actions button and select **Edit**.



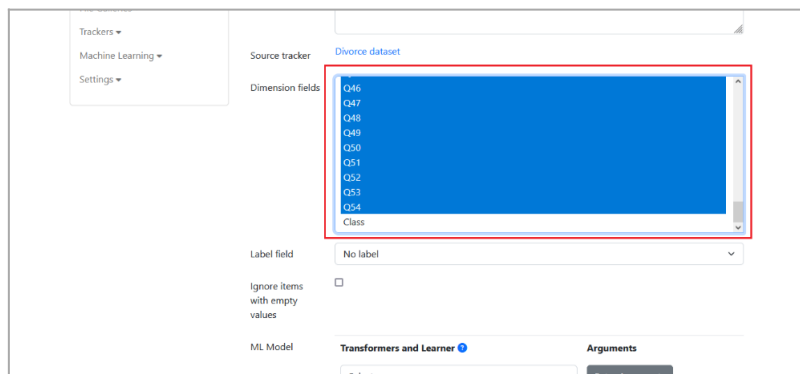
Find edit option in model's action menu



Model configuration page

## Selecting Dimension and Label Fields

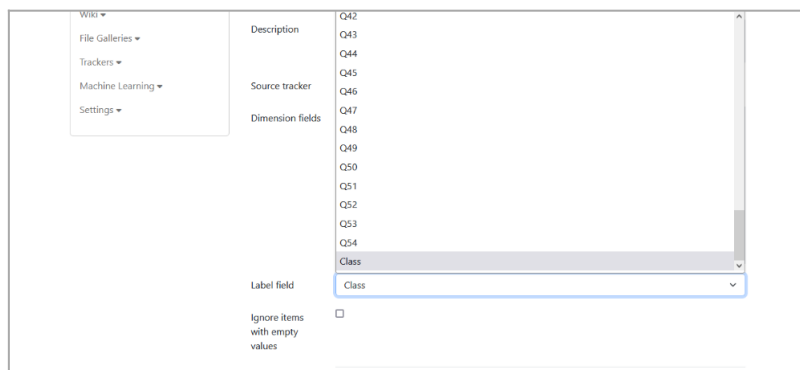
Dimension fields are chosen from a list of fields gotten from the data source tracker. These are shown in a multiselect list interface. Select a field by clicking on it. Select multiple fields by holding down the **Ctrl** keyboard key and clicking on the fields.



Select dimension fields from multi-select list

Chosen dimension fields are the data attributes that model will be trained on. Tiki will leave out all unselected fields.

The label field is the data attribute that contains the target to be predicted. A label field is required if the chosen learner is a classifier.

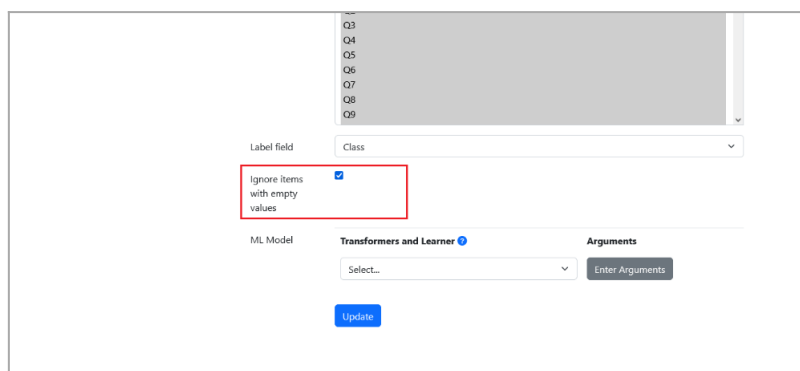


Set label field if required by learner

Some regression-based learners like [Gradient Boost](#) will also require a label field specified. In such a case, the data attribute chosen as the label field is usually expected to be of numeric type.

## Handling Empty Data Values

Before a sample is used for training, Tiki by default will replace empty numeric fields with 0. Empty categorical fields will remain as empty strings. If you do not want this behaviour, you can make Tiki to simply ignore samples with empty fields by checking the **Ignore items with empty values** option.



Check the box to ignore empty data values

With this option checked, Tiki will skip any item that contain empty fields during model training, and it will not be used to train the model.

## Adding Transformers and Learners

You use transformers to preprocess data before model training. A learner is a machine learning algorithm on which the machine learning model will be based. The type of transformers and learner you choose will depend on the structure and format of the training data and the type of target that you want to predict.

Q2  
Q3  
Q4  
Q5  
Q6  
Q7  
Q8  
Q9

Label field: Class

Ignore items with empty values: ☒

ML Model: **Transformers and Learner** ?

Select...

Enter Arguments

Update

Choose a transformer or learner

Select...

**Classifiers**

- AdaBoost
- ClassificationTree
- ExtraTreeClassifier
- GaussianNB
- KDNeighbors
- KNearestNeighbors**
- LogitBoost
- LogisticRegression
- MultiLayerPerceptron
- NaiveBayes
- RadiusNeighbors

Label field: Class

Ignore items with empty values: ☒

ML Model: **Transformers and Learner** ?

Select...

Enter Arguments

Update

Pick a transformer or learner from the list

Q2  
Q3  
Q4  
Q5  
Q6  
Q7  
Q8  
Q9

Label field: Class

Ignore items with empty values: ☒

ML Model: **Transformers and Learner** ?

KNearestNeighbors

Enter Arguments

Update

Click Enter Arguments to show popup

Add a transformer or a learner by simply selecting it from the dropdown list and clicking on **Enter Arguments** button.

KNearestNeighbors arguments

k (int): 3

weighted (bool):

kernel (Rubik's ML \Kernels\Distance \Distance): Default

Close Submit

Label field: Class

Ignore items with empty values: ☒

ML Model: **Transformers and Learner** ?

KNearestNeighbors

Enter Arguments

Update

Enter arguments

This screenshot shows a portion of the 'Edit Machine Learning Model' interface. At the top, there is a list of dimension fields (Q1-Q9) with Q9 selected. Below this, the 'Label field' is set to 'Class'. The 'Ignore items with empty values' checkbox is checked. The 'ML Model' section shows 'Transformers and Learner' with 'KNearestNeighbors' selected. The 'Arguments' section shows 'K Nearest Neighbors (k: 3, weighted: false, kernel: Euclidean)'. An 'Update' button is at the bottom.

Learner added

A popup will be displayed for you to enter the argument values to be used internally to control the transformer or learning algorithm. Tiki will autofill any left out parameters with default values.

This screenshot shows the full 'Edit Machine Learning Model' interface. The 'Name' field is 'Divorce Predictor' and the 'Description' is 'Predict whether a couple will stay married or get divorced.' The 'Source tracker' is 'Divorce dataset'. The 'Dimension fields' list includes Q1-Q9. The 'Label field' is 'Class'. The 'Ignore items with empty values' checkbox is checked. The 'ML Model' section shows 'Transformers and Learner' with 'KNearestNeighbors' selected. The 'Arguments' section shows 'K Nearest Neighbors (k: 3, weighted: false, kernel: Euclidean)'. An 'Update' button is at the bottom. A 'Show PHP error messages' button is at the bottom left.

Fully configured

This screenshot shows the 'Machine Learning Models' list page. A green success message at the top states: 'Success. Model was updated successfully. You might want to train against the source dataset.' Below the message is a table of models:

ID	Name	Description	Source
1	Divorce Predictor	Predict whether a couple will stay married or get divorced.	Divorce dataset

Success message after configuration

You add transformers in the order in which you want the data processed and you can add as many transformers as you deem fit. As a convention, the learner should be added last and only one learner is required. Adding multiple learners might result in unexpected behaviour.

Tiki internally uses [Rubix ML](#) for its Machine Learning functionality, so only transformers and learners available in Rubix ML are supported by Tiki.

Due to Tiki Tracker's robust nature, some data transformations might not be necessary. For example, [Numeric String Converter](#) works by converting all numeric values that have been given as categorical values to their equivalent integer and floating point types. Tiki will handle this automatically if the given values belong to a numeric field type in the source tracker. Applying the least possible number of transformers will help reduce model latency.

## Related links

- [Machine Learning](#)
- [Preparing Machine Learning Dataset](#)
- [Creating Machine Learning Models](#)
- [Training Machine Learning Models](#)
- [Using Machine Learning Models](#)
- [Rubix ML](#)