



TIBCO® Patterns

Learn UI Guide

Version 6.1.2 | June 2024

Contents

Contents	2
Preparing to Train a Learn Model	4
Introduction	4
Process Flow	5
Prerequisites	6
Creating, Saving, and Opening a Project	7
Creating a Project	7
Saving a Project	8
Opening a Project	9
Configuring the Data File and Its Fields	10
Reviewing List of Fields	11
Defining Features	12
Categories of Features	13
Identifying Record Pairs	21
Reviewing Record Pairs	27
Feature Scores	28
Filtering and sorting record pairs	28
Back	31
Reset	31
Training a Learn Model	32
Training a Learn Model	32
Model and Training Options	36
Small Learn Models	36
Adaptive Parameters for Model Creation	36
Setting the Model and Training Options	37
Processing Training Suggestions	41

Determining if a Model is Well Trained	41
Handling Suggestions	41
Retraining a Model	42
Types of Training Suggestions	42
Suggestions to Add Pairs to Specific Subsets	43
Adding Pairs to Subsets that Have Validation Pairs but No Training Pairs	43
Adding Pairs to Underrepresented Subsets	44
Adding Pairs to Subsets that Have Too Few True/False Labels	44
Adding Pairs to Subsets that are Found in Data File but Have No Pairs	44
Processing a Suggestion to Add Pairs	45
Filtering Record Pairs for the Suggestion Subset	45
Suggestions to Review Existing Record Pairs	46
Reviewing Possibly Mislabeled Pairs	46
Reviewing Contradictory Pairs	46
Processing a Suggestion to Review Pairs	46
Working with Trained Models	47
Trained Model	47
Testing a Trained Model	48
Setting the Query Cutoff Score	49
Saving and Reviewing Trained Models	50
Saving a Copy of Current Model	50
Reviewing Results of a Trained Model	50
Exporting a Model	51
TIBCO Documentation and Support Services	54
Legal and Third-Party Notices	55

Preparing to Train a Learn Model

This section introduces you to the concepts of machine learning models and the Learn User Interface application (Learn UI) that is used to train Learn Models in TIBCO® Patterns. It describes the required steps before the model can be trained.

Introduction

The Machine Learning Platform in TIBCO Patterns uses a machine learning model to make “Yes” or “No” decisions for problems that can be characterized by a particular set of features. In the context of the Machine Learning Platform, a feature is any characteristic of a record matching problem where the match of two data items can be quantified as a real value in the range 0.0 - 1.0.

Where :

- The value 0.0 represents the “most false” condition for the feature.
- The value 1.0 represents the “most true” condition for the feature.
- The numbers in between these values represent proportional degrees of "true" and "false". A larger value is always associated with a more positive human judgment for the feature, or at least an unchanged judgment, but cannot be associated with a decreased judgment.

Traditionally these decisions are made based on a set of manually created rules. The rule sets needed to achieve good results are often large and complex. It can be difficult to understand all of the consequences of changing or adding rules to the rule set. Using a Learn model avoids these problems. A Learn model needs to be trained in order to establish relationships between pairs of records with “True” or “False” labels. The trainer of this model can be anybody who can judge the relationships from the given examples.

The trained model can be used to predict the “True” or “False” labels of novel examples. This eliminates the need for creating an explicit rule set.

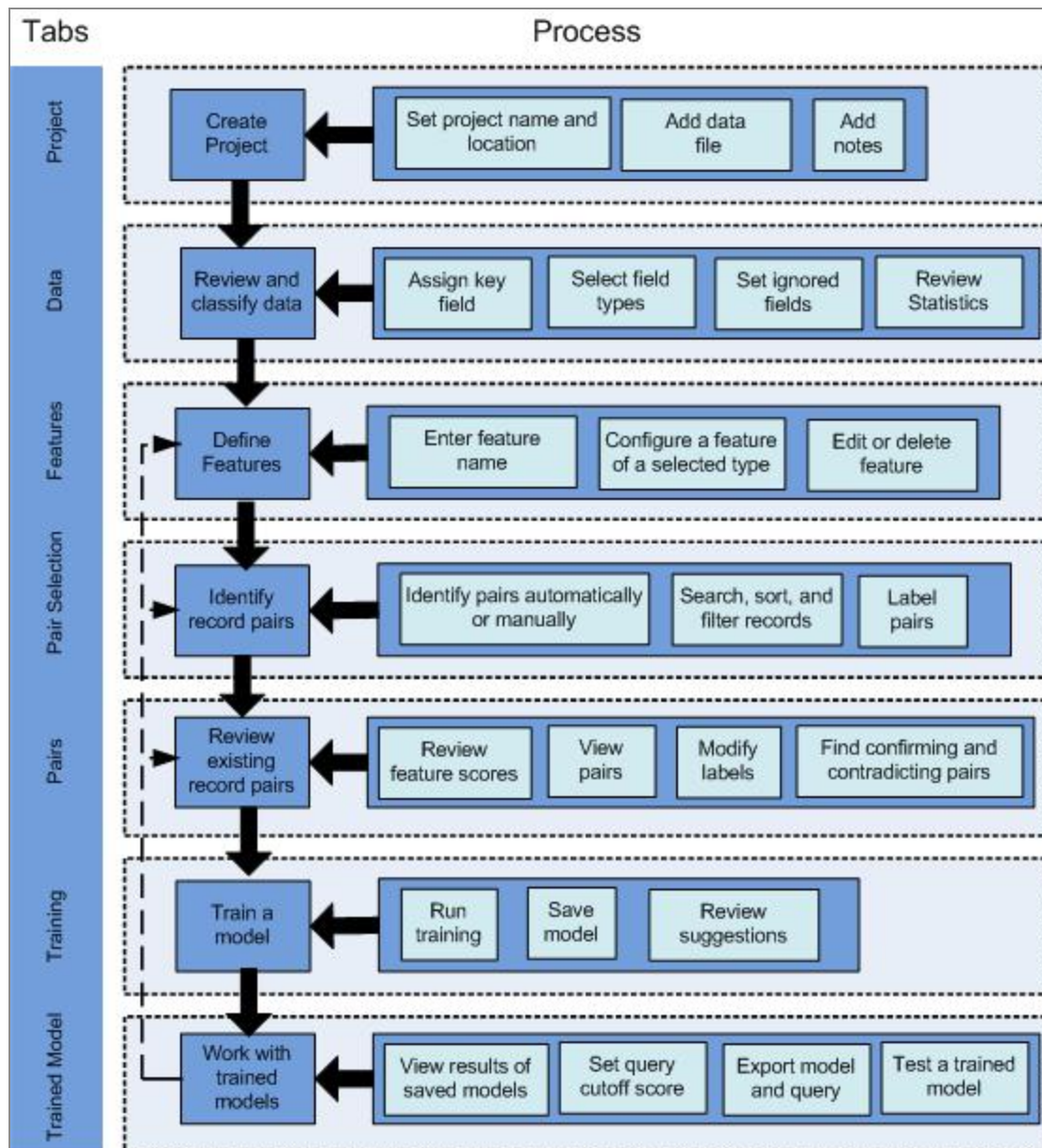
You can use Learn User Interface application to conveniently accomplish all the steps of training and evaluating Learn Models in TIBCO Patterns. The models are trained to predict whether any two records in a data table match or do not match. The Learn UI application guides you through preparing the data table, defining features, automatically finding useful record pairs for model training, labeling the record pairs and ensuring consistent labels, performing model training and evaluation, and augmenting the training data to improve model predictions.

Process Flow

The following tasks are used for creating a Learn Model by using the Learn UI application:

Task	Operation
Creating, Saving, and Opening a Project	Create a new project and add a data file to the project.
Reviewing List of Fields	Select the key field and assign field types. The statistics and distribution of values displayed by the Learn UI can determine the appropriate field type and the importance of the field in matching.
Defining Features	The set of features provides all the information on which the model bases its decisions. Each feature is a comparison of particular field values in a record pair.
Identifying Record Pairs	This step involves creating the datasets used to train and validate the model by finding record pairs that represent many possible data combinations and assigning “True” or “False” labels to the pairs. Useful record pairs can be found automatically.
Training a Learn Model	The model is trained using a number of iterations. Statistics of the trained model are displayed. The trained model can be saved together with model scores for each record pair. Features and pairs can be modified based on the provided suggestions.
Testing a Trained Model	This provides the ability to evaluate the accuracy of a trained model with record pairs that were saved with the current or another model or project.
Saving and Reviewing Trained Models	All saved models are displayed. You can export the trained model to be loaded to a TIBCO Patterns server. It can be used to find matching records in a TIBCO® Patterns search table that has the same structure as the training data.

Figure 1: Learn UI Process Flow



Prerequisites

To work with the Learn UI you must have a sample data file of the records to be matched.

The data file must have the following qualities:

- Be in a standard comma-separated values (CSV) file format.
- Contain a header line that gives the names for each field in the data file.

- Have a key field. The key field must have a unique value for each record in the data file.
- Provide a representative sample of the data to be processed.
- A practical file size limit is about 1 million records.
- The data file must have at least 100000 records to efficiently use the Low Confidence Pair Finder.

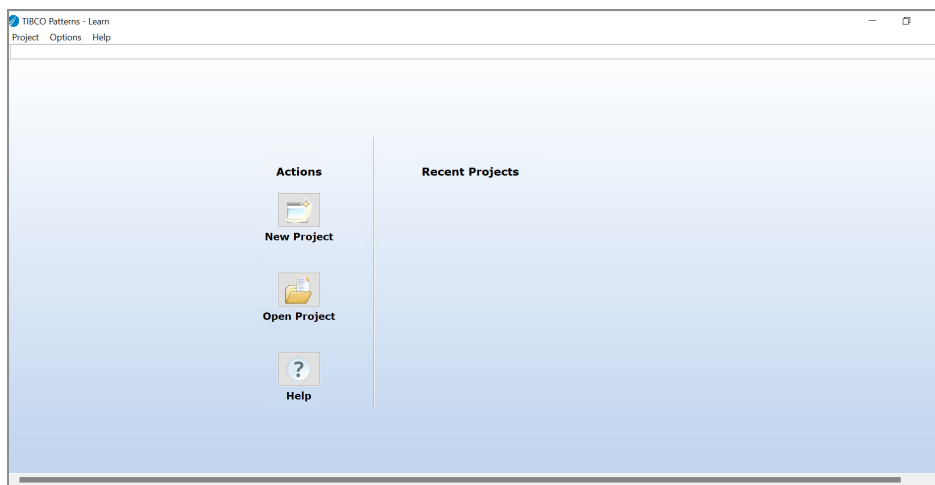
Creating, Saving, and Opening a Project

Creating a Project

This section describes how to create a new project in the TIBCO Patterns Learn UI application. To create a new project follow the steps given below:

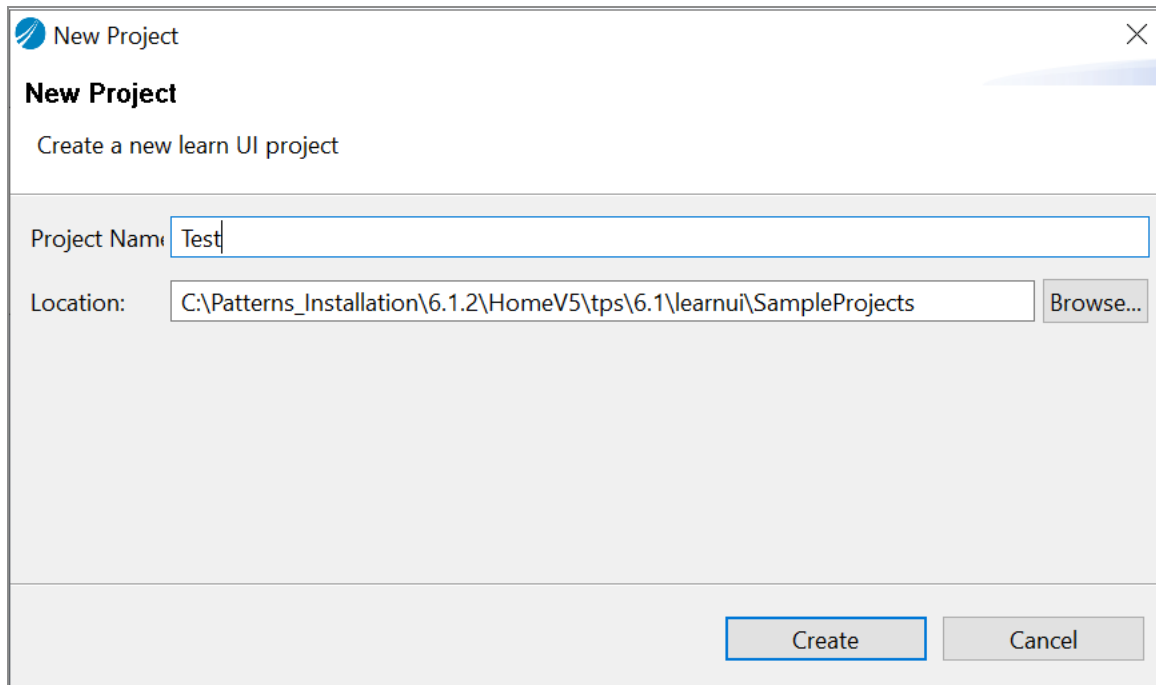
1. From the Project menu, select **New**, or click **New Project** on the application start screen.

Figure 2: Start Screen



2. Assign a name to the project. A project folder is created with this name. All files related to this project are saved in the project directory.
3. Select a location on the local drive for the project folder.

Figure 3: Create New Project



New Project

Create a new learn UI project

Project Name:

Location:

4. Click **Create** to create and save the new project. The **Project** tab is displayed.
5. Select the data file for the project. See [Configuring the Data File and Its Fields](#) for more information.
6. On the Project tab you can add or update project notes at any time.

Saving a Project

This section describes how to save a modified project in the Learn UI application. After updating a project, you can save the project in its current location or use the Save As function to save it in a new location.

To Save:

- Click **Project > Save** to save the project in its current location.

To Save As:

1. Click **Project > Save As** to save a copy of the current project.
2. Enter a new project name.

3. Click Browse to choose a new location for the project.
4. Click Save.


Opening a Project

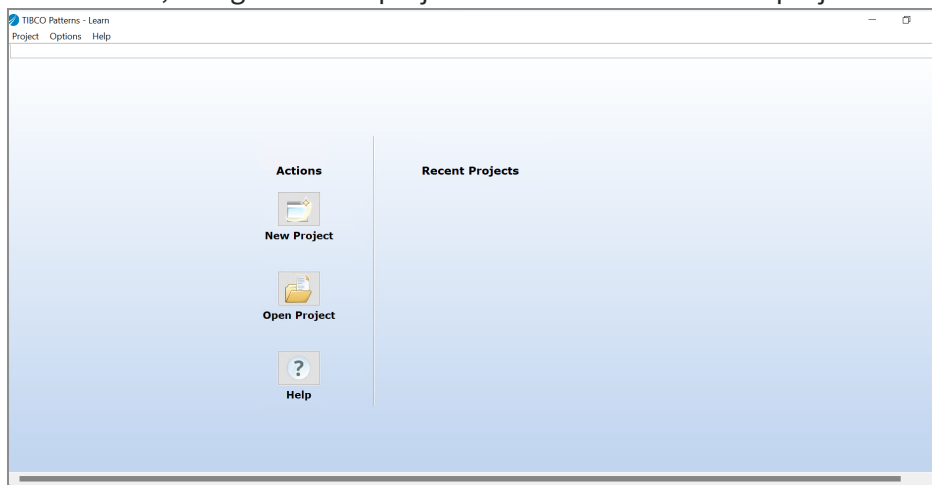
This section describes how to open a project.

A project can be opened in one of the following ways:

Open a Project

This way is most useful if the project you want to open was not recently opened.

1. Click the Open Project icon  or click the menu item Project > Open.
2. Click Browse, navigate to the project location and select the project folder.



3. Click Open

Open a Recent Project

4. Click the *project name* in the Recent Projects list.
- Or
5. Click Project > Open Recent and select one of the recent projects from the sub-menu.

Suggested Next Actions for the Project

On the Project tab, a list of Suggested actions is provided. The items suggest the next steps that should be taken for this project. Click on any item in the list to perform the described action. The suggestions depend on the current state of the project. These suggestions are often based on the information missing from the project.

For example:

- If the key field was not assigned to the data file, you are asked to do so.
- If some pairs were left unlabeled, you are asked to label them.
- If no pairs or too few pairs were added, you are asked to add more pairs.
- If some pairs have been marked for review, you are prompted to review them.

Configuring the Data File and Its Fields

The data file must be in CSV format. The first row of the file must contain field names.

The data file should contain a representative sample of records in the possibly much larger table on the TIBCO Patterns server where the trained model is eventually used. The data file should have enough records to create a large variety of record pairs. It should have at least 100000 records to use the Low Confidence Pair Finder efficiently. An empty data file should not be assigned, otherwise you will not be able to create any record pairs or train a model.

1. On the Project tab, click Assign.
2. Click Browse and locate the data file. The Learn UI provides a choice to associate the selected data file with the project in one of the following ways:

From the project directory

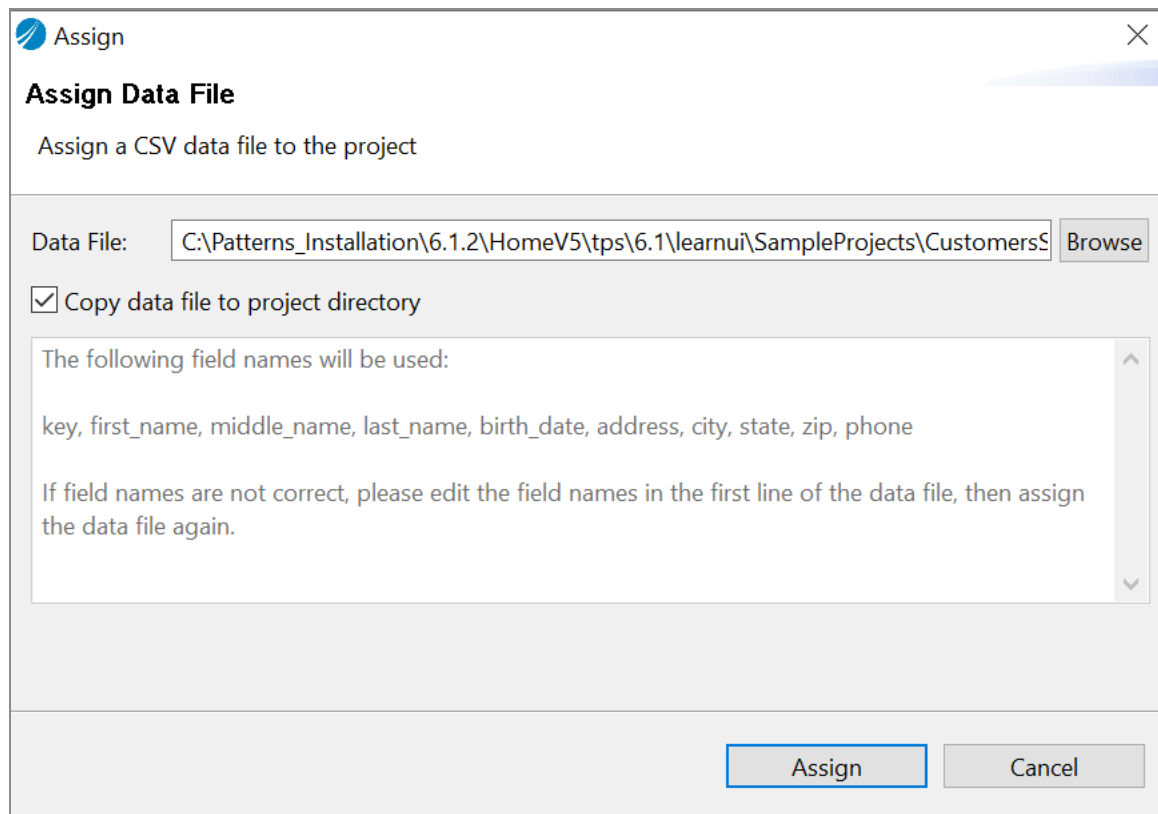
Select Copy data file to project directory. In this case, the file is copied to the project directory. The copied file is accessed by the project. This makes the project folder completely portable so that it can be easily copied to another computer.

From a different location

Do not select Copy data file to project directory. In this case, the file is linked to the project without copying it. This can be used to create several Learn projects on the same system that use the same data file, without creating multiple copies of the data file.

Warning: Changes to the data file can invalidate the entire project or the labels of existing pairs. Therefore, when using this option, you must take care that the data file is not modified, renamed, or deleted as long as the Learn UI project exists. In case you make changes to the field values in the CSV file that do not make any existing pair labels invalid, then you can still use such data file. The application detects data file changes and suggests to automatically update field data in the existing pairs from the modified CSV file.

Figure 4: Assign Data File



The image shows a dialog box titled "Assign" with a close button (X) in the top right corner. The main heading is "Assign Data File" and the subtitle is "Assign a CSV data file to the project". Below this, there is a "Data File:" label followed by a text input field containing the path "C:\Patterns_Installation\6.1.2\HomeV5\tps\6.1\learnui\SampleProjects\CustomersS" and a "Browse" button. A checkbox labeled "Copy data file to project directory" is checked. Below the checkbox is a scrollable text area containing the text: "The following field names will be used: key, first_name, middle_name, last_name, birth_date, address, city, state, zip, phone". Below this text is a note: "If field names are not correct, please edit the field names in the first line of the data file, then assign the data file again." At the bottom right of the dialog are two buttons: "Assign" and "Cancel".

Reviewing List of Fields

After the data file is assigned, the list of fields from the file is displayed on the Data tab. Statistics for all fields are also displayed. These statistics might help determine the appropriate field type and also whether a field is useful in determining a record match.

Figure 5: Reviewing the List of Fields

You can perform the following operations in this tab:

3. Select key field

Choose the key field for the data table in the Key column. The field selected must contain a unique value for every record.

4. **Change field type**

The default field type is Searchable Text. You can change this by clicking the field type. Then choose the new field type from the drop-down list. Fields that contain date values, for example, Date of Birth, generally should be changed to Date or Searchable Date type. Fields to be compared as numbers, such as size or weight, should be assigned the Integer or Floating Point field type. However, numeric ID fields, like order numbers, phone numbers, and ZIP codes are best left as Searchable Text fields to be compared as text.

Statistics for a Searchable Date field calculated on the Data tab, must be identical to statistics when the Date field type is selected for the same field.

The custom filter that is applied on the Pair Selection tab for the field of type Date must be preserved when the field type is changed to Searchable Date, and vice versa.

5. **Ignore fields**

Selecting the Ignore checkbox for a field eliminates this field from the pair selection and learning process. Fields that are never useful for matching can be marked as ignored.

These fields are not displayed in all the other tabs, avoiding clutter when examining records and pairs.

Defining Features

After the data file and field types are configured, the next step is to create features. A feature that is defined in Learn UI is a group of fields used in matching that provides one or more input feature scores for training a model.


When creating features, consider the following:

- Create features for fields that decide whether any two records match. Start with defining features for the fields that are the most important. It is not necessary for each table field to be included in one or more features.
- There is a limit on the total number of model features. The current total appears at the bottom of the Features tab. In typical systems, the practical limit is 10 or 11 model features. Using more features can result in very large model files that are likely to require too much memory.
- A separate feature for each field is preferred to combining them into a single feature. Because the number of features is limited, you can combine several related fields into one feature to stay within this limit.

- If fields must be combined into one feature, select fields that make up parts of a whole, such as the components of an address.
- Consider removing any features defined for fields that provide very little additional information. For example, remove the feature for a State field if more than 90% of records are from the same state.
- Metadata fields, such as record creation date, the person who last updated the record, are typically not useful for record matching and must not be used to create features.
- A higher match score for a feature must always represent a higher (or unchanged, but never lower) probability of a positive match for the record pair as a whole, provided that the scores for all other features remain the same.

To add a new feature:

1. Open the Features tab.
2. If some features have already been created, click Add Feature. The Add Feature screen is displayed. If no features have been created, that screen is displayed immediately.
3. Specify the feature name. Every feature must have a unique name.
4. Select the feature category.
5. Select the feature type.
6. Click Next.
7. Select one or more fields to be used in the feature and configure the options displayed for the selected feature type.
8. Click Finish.

 **Note:** You cannot create a new feature unless a key field for the data table has been selected.

The features are displayed in a table in the Features tab.

Categories of Features

Features are of the following categories:

- Generic
- Data Specific

TIBCO Patterns - Learn: CustomersSample

Project Options Help

Project Data Features Pair Selection Pairs Training

Add Feature

Feature Name:

1. Select Feature Category:

- Generic
- Data-specific

Description:
Generic features can be used to create any feature for general purpose applications, especially when a data-specific feature is not defined for your application area.

2. Select Feature Type:

- Simple
- Cognate
- Date
- Predicate

Description:
A Predicate feature uses exact matching to evaluate a predicate expression over one or more field values.

Next Cancel

Figure 6: Selecting Feature Category and Type

Generic Features

Generic features can be used to create any feature for general purpose applications, especially when a data-specific feature is not defined for your application area. Generic features correspond to the basic query types available in TIBCO® Patterns. For more information on queries, see the section "Designing Queries for Patterns" in the TIBCO® Patterns Concepts guide.

Types of Generic Features:

- **Simple**

A Simple feature compares one or more fields in two records. A single text string constructed from the selected field values in one record is compared to the concatenated value of the selected fields in the other record. The feature score reflects the contributions of the whole or partial matches found across the selected fields. You can add a thesaurus file and specify a thesaurus type and weight for a simple feature. If the Match Empty Values checkbox is selected, and all the selected fields are empty in both records in the record pair, the feature score is 1.0 instead of the empty score -1.0.

Figure 7: Simple Feature

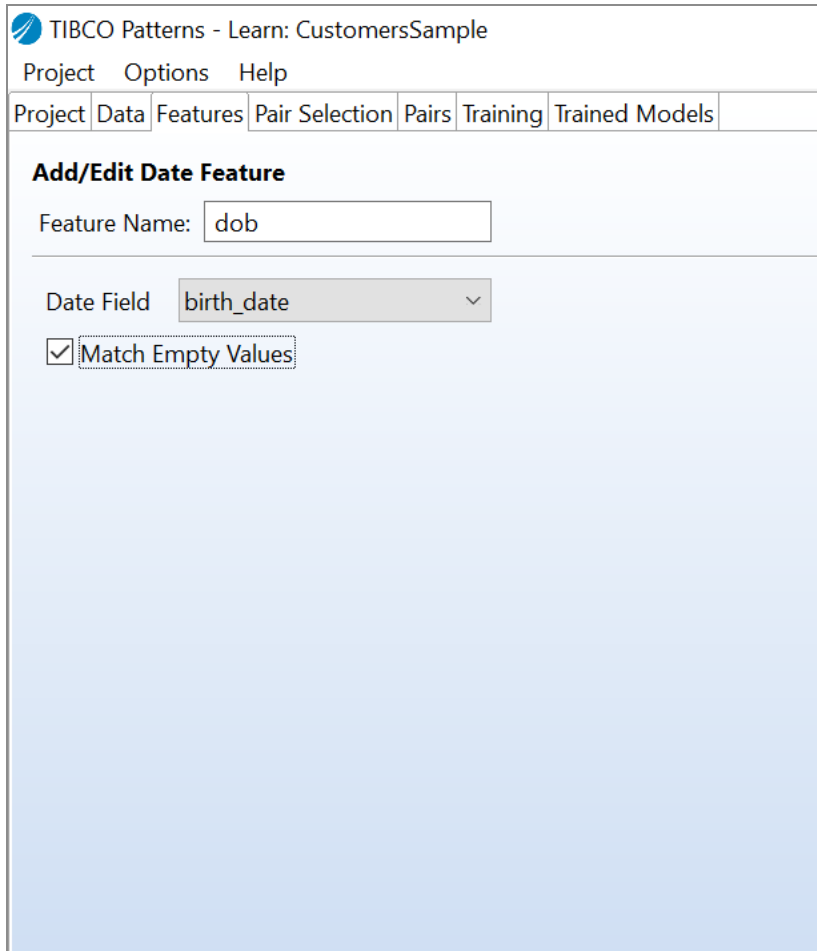
The screenshot shows the 'Simple Feature' configuration window in TIBCO Patterns. The window title is 'TIBCO Patterns - Learn: CustomersSample'. The menu bar includes 'Project', 'Options', and 'Help'. The breadcrumb trail is 'Project > Data > Features > Pair Selection > Pairs > Training'. The 'Simple Feature' section has a 'Feature Name' field containing 'Address'. Below this is a table with columns 'Field', 'Included in Featu...', and 'Weight'. The table lists fields: first_name, middle_name, last_name, address, city, state, zip, and phone. The 'address' row is selected, with a checked box in the 'Included' column and a weight of '1.0'. To the right of the table, there is a 'Thesaurus File' field with a 'Browse...' button, a 'Thesaurus Type' dropdown menu set to 'Substitution', a 'Thesaurus Weight' field set to '1.0', and a 'Match Empty Values' checkbox which is unchecked. At the bottom right, there are 'Finish' and 'Back' buttons.

Field	Included in Featu...	Weight
first_name	<input type="checkbox"/>	
middle_name	<input type="checkbox"/>	
last_name	<input type="checkbox"/>	
address	<input checked="" type="checkbox"/>	1.0
city	<input type="checkbox"/>	
state	<input type="checkbox"/>	
zip	<input type="checkbox"/>	
phone	<input type="checkbox"/>	

- **Cognate**

A Cognate feature specifies a structured comparison over a group of fields. It is able to match a value even if it is entered into a wrong field. Use cognate features for closely related fields subject to frequent misfielding. For example, the fields `first_name`, `middle_name`, and `last_name` can be combined by using a cognate feature. You can add a thesaurus file for a cognate feature and specify the thesaurus type and weight. In addition, you can specify a noncognate weight (score penalty if a value is entered into a wrong field), and an empty field penalty (a score penalty to discount unmatched data that can be attributed to an empty field in either of the two records in a record pair). If the Match Empty Values checkbox is selected, and all the selected fields are empty in both records in the record pair, the feature score is 1.0 instead of the empty score -1.0.

Figure 9: Date Feature



TIBCO Patterns - Learn: CustomersSample

Project Options Help

Project Data Features Pair Selection Pairs Training Trained Models

Add/Edit Date Feature

Feature Name:

Date Field

☒ Match Empty Values

- **Predicate**

A Predicate feature uses an exact matching predicate expression to compute the feature score. These expressions are defined using the language for the TIBCO Patterns predicate expressions. A Predicate feature requires you to compose a valid predicate expression. For more information about constructing predicate expressions, see the sections "Constructing Predicate Expressions" and "Predicate Queries" in TIBCO® Patterns Concepts Guide.

Predicate expressions, as described in the Concepts Guide, reference table record field values as "\$field-name". The predicate expressions used in the Learn UI also have an ability to reference fields in the query record. Query record field values are referenced as \${field-name}.

Note: If the query record field value is to be treated as a text value it must be enclosed in double quotes, for example "\${field-name}". For more information about using query record field values in predicate expressions, see the description of NetricsPredicateMapper class in the Java API documentation.

A predicate expression must refer to the same fields in both records in a record pair, one being the query record, the other the table record. The result of a predicate expression must be the same if the two records in any pair are switched.

Figure 10: Predicate Feature

TIBCO Patterns - Learn: CustomersSample

Project Options Help

Project Data Features Pair Selection Pairs Training

Add/Edit Predicate Feature

Feature Name:

Predicate expression:

A symmetrical predicate expression that refers to the same fields in both records in a record pair

Predicate Expressions Reference

`$(FieldName)` - value of table field `FieldName` (from one record in the pair).
`$(FieldName)` - value of query field `FieldName` (from the other record in the pair). The value is inserted into predicate expression, so a text value needs to be quoted: `"$(FieldName)"`.
 The result of a predicate expression must be the same if the records in any pair switch places.

Unary operators: `int`, `double`, `date`, `date_time`, `eudate`, `eudate_time`, `block`, `-`, `+`, `not`, `split`, `abs`.
 Binary operators: `+`, `-`, `*`, `/`, `**`, `and`, `or`, `=`, `~`, `<`, `<=`, `>`, `>=`, `~<`, `~>`, `~<=`, `~>=`, `in`, `i_in`, `superset`, `subset`, `split` (using a specified separator). `"~"` means case-insensitive.
 Functions: `geodistance`, `if`, `toscore`. `{p1, p2, ..., pn}` - parameter list for functions.
 String constants: `"String value"`. Boolean constants: `?TRUE?`, `?FALSE?`.

See TIBCO Patterns Concepts Guide for more details.

Data-Specific Features

Data-specific features are predefined for a certain domain. Use them first if your data matches the purpose of the feature. A data-specific feature is a predefined combination of underlying model features. It eliminates the need to define the exact parameters for several generic features. The data-specific features use parameters that in most cases are best for the type of data indicated.

Types of data-specific features:

- **Person Name**

This feature compares the similarity of First Name, Last Name, and an optional Middle Name field. You can specify a thesaurus to be used in underlying model features that include the First Name or Middle Name fields.

Figure 11: Person Name Feature

The screenshot shows the 'Add/Edit Person Name Feature' configuration window in the TIBCO Patterns - Learn: CustomersSample application. The window has a title bar with 'Project', 'Options', and 'Help' menus. Below the title bar is a tabbed interface with 'Project', 'Data', 'Features', 'Pair Selection', 'Pairs', and 'Training' tabs. The 'Features' tab is active, and the 'Add/Edit Person Name Feature' dialog is open. The dialog has a 'Feature Name' field with the value 'Name'. Below this are three dropdown menus for 'First Name Field' (set to 'first_name'), 'Middle Name Field' (set to 'middle_name'), and 'Last Name Field' (set to 'last_name'). To the right of these is a 'Thesaurus File' button labeled 'Browse...'. Below the dropdowns is a 'Thesaurus Type' dropdown set to 'Substitution' and a 'Thesaurus Weight' input field set to '1.0'. At the bottom right of the dialog are 'Finish' and 'Back' buttons.

- **Gender Feature**

This feature determines whether a gender field in two records has the same meaning. It also allows the model to predict differently for Male and Female record pairs. The codes used to indicate male and female are defined in this feature. These gender codes are selected from the list of the most frequent field values.



Note: Gender codes are not case sensitive.

Figure 12: Gender Feature

[illegible]

The list of created Features.

Figure 13: Pair Selection

The screenshot shows the TIBCO Patterns - Learn: CustomersSample application. The main window displays a table with 249941 records. The table has columns: key, first_name, middle_name, last_name, birth_date, address, city, state, zip, and phone. Two records are highlighted in blue: record 200100 (ETHEL, MURIEL, 03/25/1970, 1019 ELIZABETH BLACKWELL S, EATONTOWN, PAA, 79036, 5305225621) and record 200101 (ELIJAH, SAITTA, 12/24/1935, 846 W WINNCHESHER RD, BRANY ACAMP, NNJ, 517, 88267647710).

Below the table, the 'Pair Selection' panel is visible. It includes a 'Suggest Pairs' button and an 'Add Record to Pair' button. The 'Selected Record Pair' section shows two records selected: record 0 (ETHEL, MURIEL, 02/25/1971, 1019 ELIZABETH BLACKWELL S, EATONTOWN, PAA, 79036, 5305225621) and record 200100 (ETHEL, MURIEL, 03/25/1970, 1019 ELIZABETH BLACKWELL S, EATONTOWN, PAA, 79036, 5305225621). The 'Label' section has three radio buttons: 'True' (selected), 'False', and 'Unsur'. There are also 'Save Pair' and 'Mark as Always False' buttons.

Selecting Pairs Manually

You can create record pairs manually by first selecting two records and then setting the appropriate label for the pair.

1. Select a record and click Add Record to Pair. Repeat this step to form an appropriate pair.

Note: You can use various [Supporting Functions](#) like searching and sorting records which make them easier to locate.

2. Select an appropriate label from the available options. To assign pair labels separately from selecting pair records, skip step 2.
3. Click Save Pair.

When you click the Save Pair button, the selected pair is randomly added to either the Training dataset or the Validation dataset. The Training dataset is used to train the model. The Validation dataset is used to monitor the performance of the trained model over unseen record pairs and to stop the training process when the validation error rate is the lowest.

Labels

- **True**

Select the True label if the two records match (represent the same entity).

- **False**

Select the False label if the two records do not match (represent different entities).

- **Unsure**

Select the Unsure label if you are not sure whether the records match or not. Record pairs with Unsure labels or without any assigned label do not participate in the learning process, but you can change the label later.

Eventually all pairs are expected to have a label to participate in the learning process.

Finding Useful Pairs Automatically

You can also find record pairs automatically by using Suggest Pairs button on the Pair Selection tab.

The Suggest Pairs button starts the Low Confidence Pair Finder, which searches for new pairs in the background. You can review and label the already found pairs while the search is running. The pairs that are found are useful for training of the Learn model, because they address situations that were not sufficiently trained earlier. Thus, the existing datasets can be augmented to cover new matching scenarios or a new model can be trained from the very beginning by using only the automatically found pairs.

The pairs are found by examining records in the existing table. The confidence of model predictions is used to determine the pairs that are likely to be useful and indicate how reliable is the prediction.

The confidence of the model prediction is determined by the similar record pairs, which were used during model training as follows:

- If the model has never seen similar pairs during training or it has seen similar pairs with contradictory labels, the confidence is low.
- If the model has seen many similar pairs with consistent labels during training, the confidence is high.

The Low Confidence Pair Finder focuses on finding pairs with the lowest confidence. After you label a pair and add it to the Training dataset, the retrained model is likely to predict this pair with an increased confidence.

You must assign a True, False, or Unsure label to the found record pair before saving it. You can also mark the subset represented by the record pair as Always False (for more information, see the section [Always False Subsets](#)). Once the pair is labeled and saved, the next found pair is automatically displayed for labeling.

To stop automatically finding pairs, click the Stop Suggesting button, then label and save the last found pair that is displayed.

An existing Learn model is required to find pairs automatically. You can train an initial model even when no pairs have been saved (for more information, see the section [Training a Learn Model](#)), and then click the Suggest Pairs button. Alternatively, you can simply click the Suggest Pairs button and the application offers to train and save the initial model.

The number of the found pairs is displayed on the Pair Selection tab. The Low Confidence Pair Finder tends to find a large number of record pairs when no pairs or just a small number of pairs has been used to train the model. It tends to find fewer pairs when many pairs were already used to train the model. If very few pairs are found, it is recommended to wait for several minutes or longer to see if a larger number of pairs is found. Eventually the process no longer finds any record pairs within a reasonable time, which means that the model is already well trained for the given data table. To ensure that a sufficient number of pairs can be found in a reasonable time, the data table should contain a representative sample of at least 100000 records.

While finding pairs automatically, the system periodically offers to retrain the model. It is recommended to do this since the model learns the new matching situations which reduces the total number of pairs that you need to label. After the model is automatically retrained and saved, you can review the training results and then click the Suggest Pairs button again to continue the process.



Note: You cannot change tabs while pairs are being suggested. To stop suggesting pairs click Stop Suggesting and change the tab as required.

Subsets

When field values for a certain feature are missing, the criteria used to determine a record match are likely to be different. For example, if a Social Security number is missing, a match on a secondary field, such as birth date or address, is likely to be crucial to establish a match of the two records, whereas if the Social Security number is present, the secondary fields might be almost irrelevant. Therefore, the model learns differently depending on what feature scores are present or absent, and must be trained for each case.

A subset defines which feature scores are present and which are absent. There is a separate subset for each combination of present and absent feature scores. If a Simple or Cognate feature uses multiple fields, its score is absent only if all the fields used are empty in either record of the pair.



Note: Exception: the score of a Predicate feature is absent if at least one field used in the predicate expression is empty or invalid.

Always False Subsets

Some subsets of present feature scores can be marked as Always False. Record pairs that belong to these subsets or any of their subsets are always classified as False by the Learn model. Even when the present feature scores in such record pair represent an exact match in the two records of the pair, this information is still not sufficient to classify the pair as a True match.

For example, in a Learn model for person matching, having two records that only have the City and State features that match exactly (all other feature scores are empty) is not sufficient to determine that the two records represent the same person (there are many people living in the same city and state). Thus a subset that only has non-empty City and State feature scores can be marked as Always False. This causes any record pairs that have only a non-empty City feature score, or only a non-empty State feature score, or that have all empty feature scores to also be classified as False, since these are subsets of the original subset that was marked as Always False.

You can use the Always False subsets to make the decision about the subset only once instead of labeling a potentially large number of pairs for the same subset, or trying to find two records with exact matches for the features in this subset to demonstrate that the exact match must still be classified as False. Also having Always False subsets reduces the size of the model file and speeds up model predictions for these subsets.

To mark a subset as Always False, select a pair of records that represent that subset, select the False label and then click the Mark as Always False button. Review the list of present feature scores on the confirmation dialog and confirm the Always False subset. You can also mark an automatically found pair as Always False.

The record pairs that represent Always False subsets are stored in a separate Always False Subsets dataset. You can review these pairs in the Pairs tab. Deleting such pair removes the Always False subset.

A Learn model assigns a score of 0.0 and a confidence of 1.0 for all pairs that belong to one of the saved Always False subsets or their subsets.

Supporting Functions

- **Basic search**

This function searches for the string specified in the search text-box. The basic search function finds the search string anywhere in the text of any field. It does not search for whole words.

- **Reset**

The Reset button is used to reset all changes made to the record organization on the Pair Selection tab. It removes all sorting orders, searches and filters for the data table. This will make the records go back to the original order in the CSV file, and all records will be shown. In addition, any automatic filters used to process model training suggestions will be removed.

Column Context Menu Functions

The following functions can be accessed by right-clicking on column title:

- **Sort**

Using sorting, you can find records with the same or similar values in the sorted fields. Sorting each field makes it easy to analyze it individually. Each field can be sorted in ascending or descending order. Multiple columns can be selected for sorting. The last column that is sorted becomes the primary sort column.

- **Clear Filter**

This function removes any filter that is currently applied for the selected field. It does not remove any filters for other fields.

- **Filtering blank and non-blank field values**

Filtering by blank and non-blank field values can be used to view subsets of present and empty field values. It helps to narrow down the list of records and focus on a specific subset of present field values.

- **Custom Filter**

Select the Custom Filter menu item to specify up to two custom filters for the selected field. If both filters are specified, they can be combined with an AND or OR operation. To specify each filter, select a filter type from the drop-down menu and enter a value to be used by the filter. The types of custom filters available in the drop-down depend on the type of the selected field.

- **Hide Field**

This function makes the selected field invisible in the Pair Selection tab. If you want to show a hidden field again, use the Show/Hide Fields item in the Table Functions drop-down menu.

Table Functions

The following functions can be accessed through the Table Functions drop-down menu:

- **Advanced Search**

This function provides an additional search functionality that is field-specific. There are several search operations that can be specified for each field: equals, contains (contains the search string anywhere in the field), and contains phrase (contains the specified whole word or a phrase of whole words).

- **Sort dialog**

This function provides an ability to precisely define and change the sorting by multiple columns. You can add a number of columns to the list of sorted fields, specify ascending or descending order for each column, and move any column up or down the list of sorted fields.

- **Clear Filter and Sort State**

After searching, filtering, or sorting the field values, you can use this function to make the records go back to the original order in CSV file and to display all records. Unlike the Reset button, this function does not remove any automatic filters that might have been applied to process model training suggestions.

- **Show/Hide Fields**

With this function, you can select any field to be hidden or shown again in the Pair Selection tab. If the field is selected, it will show in the Pair Selection tab. If you clear the checkbox, the field will be hidden from the tab. Unlike the Ignore checkbox in the Data tab, this function hides the field only from the Pair Selection tab.

Reviewing Record Pairs

All saved record pairs are listed in the Pairs tab.

Figure 14: List of Existing Pairs

TIBCO Patterns - Learn: CustomersSample

ProjectOptionsHelp

ProjectDataFeaturesPair SelectionPairsTrainingTrained Models

Record Pairs

Displaying 689 record pairs

BackDeleteDelete AllShow/Hide FieldsFilterReset

Review	Data Set	Current Label	Training Label	Model Score	Prediction	Confidence	key	first_name	middle_name	last_name	birth_date	address	city	state	zip	phone
<input checked="" type="checkbox"/>	Train	True	True	0.976	Correct	1.000	6	ETHEL		MURIEL	1971/02/25	1019 ELIZABETH BLACKWELL S	EATONTOWN	PAA	79036	53052251
							200100	ETHEL		MURIEL	1970/03/25	1019 ELIZABETH BLACKWELL S	EATONTOWN	PAA	79036	53052251
<input type="checkbox"/>	Train	False	False	0.007	Correct	1.000	7	IVA		ENGERT	1954/05/26	1515 CALIMYRNA AVE	PRAATTSVILLE	L	46795	31347301
							6	ETHEL		MURIEL	1971/02/25	1019 ELIZABETH BLACKWELL S	EATONTOWN	PAA	79036	53052251
<input type="checkbox"/>	Train	True	True	0.902	Correct	1.000	15	ELIJAH	ALAN	SAITTA	1935/11/24	846 W WINNCHESHER RD	BRANDY CAMP	NNJ	56117	78676471
							200101	ELIJAH		SAITTA	1935/12/24	846 W WINNCHESHER RD	BRANDY CAMP	NNJ	517	88267641
<input type="checkbox"/>	Train	False	False	0.008	Correct	1.000	16	ENRIQUETA	DESTINY	ZYGADLO	1991/06/15	1281 W BEVERLEY ST	RADISSON	NJ	54486	40278241
							15	ELIJAH	ALAN	SAITTA	1935/11/24	846 W WINNCHESHER RD	BRANDY CAMP	NNJ	56117	78676471
<input type="checkbox"/>	Train	True	True	0.914	Correct	1.000	27		LESTER	CHENIER	1960/03/16	RR 9 BOX 45A	SNOWSHOE	MO	62932	77972141
							200102			CHENIER	1960/04/16	RR 9 BOX 45A	SNOWSHOE	MO	62932	77972141
<input type="checkbox"/>	Train	False	False	0.010	Correct	1.000	28	ANN		MALIK	1948/05/04	2282 N CHICAGO AVE	DPOCA	HI	5452	70763191
							27		LESTER	CHENIER	1960/03/16	RR 9 BOX 45A	SNOWSHOE	MO	62932	77972141
<input type="checkbox"/>	Train	True	True	0.840	Correct	1.000	29	RGELIO	ERIC	UDUSTIN	1950/12/08	1334 DELACODO AVE	CAMANNCH	FL	37544	51330171
							200103	RGELIO	ERIC	UDUSTIN	1950/01/07	1334 DELACODO AVE	CAMANNCH	FL	37544	51330171
<input type="checkbox"/>	Train	False	False	0.006	Correct	1.000	30	WAYNNE	KARL		1941/11/29	1106 STATION CREEK RD	MOGGADORE	HI	49876	31539721
							29	RGELIO	ERIC	UDUSTIN	1950/12/08	1334 DELACODO AVE	CAMANNCH	FL	37544	51330171
<input type="checkbox"/>	Train	True	True	0.800	Correct	0.862	40	NATHAN	DEAN	CLARENCE	1968/06/20	488 BIG ISLAND RD	CASTLEBERRY	IL	47338	60175111
							200104	DEN		CLARENCE	1968/07/20	488 BIG ISLAND RD	CASTLEBERRY	IL	47338	60175111
<input type="checkbox"/>	Train	False	False	0.009	Correct	1.000	41	RICARDO	LOUIS	FELKEL	1984/04/28	889 MISSION HOME RD	PEACE DALE	LA	78363	31955411
							40	NATHAN	DEAN	CLARENCE	1968/06/20	488 BIG ISLAND RD	CASTLEBERRY	IL	47338	60175111
<input type="checkbox"/>	Train	False	False	0.479	Correct	0.771	45	VIOLETTE	ROXIE	ULLOM	1984/01/12	1123 RIVIERA AVE	WESCO		75663	50830771

Feature Scores for selected Record Pair

Name: First Name	Name: Last Name	Name: Cognate Name	Address	City	Phone	Zip	dob
1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.500000

In this tab, you can perform the following functions:

- **Assign or change the Current Label**

The label of a selected pair can be changed from the drop-down list provided in the Current Label column.

- **Mark a pair for future review**

Use the checkbox in the Review column. The marked pairs can be reviewed later, even after saving and loading the project. It is convenient to use the filter function to display only the pairs that need to be reviewed.

- **Delete and Delete All**

The Delete button deletes the selected pair. The Delete All button deletes all the pairs in the project.

- **Show/Hide Fields**

You can select which fields should be displayed in the grid based on your requirements. Use the Show/Hide Fields button or the Hide Field item in the column context menu. Selecting the checkbox for the field makes it visible in the grid, and vice versa.

Feature Scores

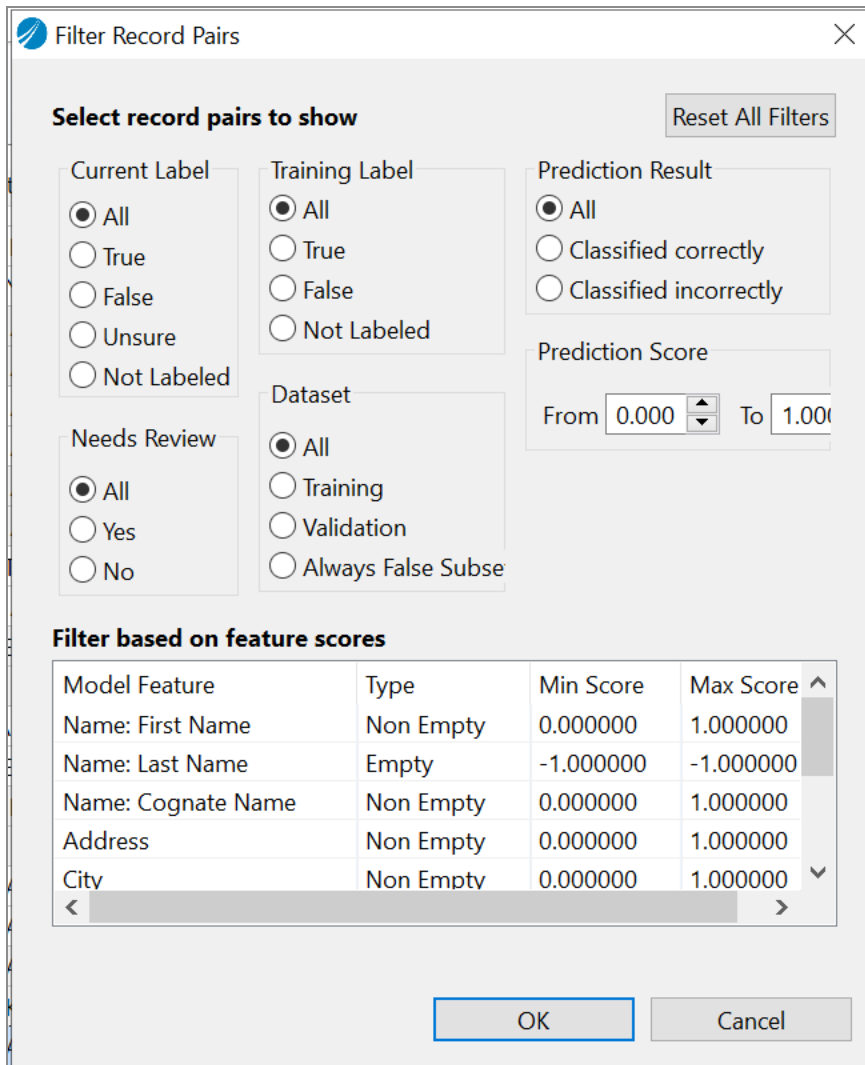
Feature scores appear only after you train a model. Feature scores for a selected record pair represent the entire input information that is used to train or validate the model with that pair. One feature score is calculated for each model feature. A feature score indicates the degree of match between the field values that are included in the model feature. It is a real number ranging from 0.0 to 1.0. A feature score can also be equal to -1.0. The meanings of feature score values are:

-1.0	Indicates that all the field values included in the feature are empty or invalid in one or both records of the pair. However, if the Match Empty Values checkbox is selected for this feature, and the field values are empty in both records, then this results in the 1.0 score. Exception: for a Predicate feature, this value indicates that at least one field used in the predicate expression is empty or invalid.
0.0	Indicates that there was no common data.
1.0	Indicates an exact match for this model feature.

Filtering and sorting record pairs

Clicking the Filter button displays a Filter Pairs dialog which can be used to filter the list of pairs that are displayed in the grid. Any active filters are indicated by filter icons on the appropriate column titles in the Record Pairs and Feature Scores grids.

Figure 15: Filter Pairs dialog



Filter Record Pairs [X]

Select record pairs to show [Reset All Filters]

Current Label

☒ All

☐ True

☐ False

☐ Unsure

☐ Not Labeled

Training Label

☒ All

☐ True

☐ False

☐ Not Labeled

Prediction Result

☒ All

☐ Classified correctly

☐ Classified incorrectly

Needs Review

☒ All

☐ Yes

☐ No

Dataset

☒ All

☐ Training

☐ Validation

☐ Always False Subse

Prediction Score

From To

Filter based on feature scores

Model Feature	Type	Min Score	Max Score
Name: First Name	Non Empty	0.000000	1.000000
Name: Last Name	Empty	-1.000000	-1.000000
Name: Cognate Name	Non Empty	0.000000	1.000000
Address	Non Empty	0.000000	1.000000
City	Non Empty	0.000000	1.000000

[OK] [Cancel]

— **Current label**

Shows pairs where the current label matches the selected item.

— **Training Label**

Shows pairs where the label at the time the model was last trained matches the selected item.

— **Latest Prediction Result**

Shows pairs where the prediction from the last training run either matches (Classified correctly) or does not match (Classified incorrectly) the Training Label.

— **Needs Review**

Shows pairs based on the “Review” checkbox in the grid.

— Dataset

Show pairs in the indicated dataset. You can see which pairs are used to train the model and which pairs are used to validate the model. You can also filter the pairs that represent Always False subsets.

— Prediction Score

Show pairs based on the model score from the latest training run. The model score is the score output by the model for this pair. By default a score of 0.5 or above is considered a match, a score less than 0.5 is considered a non-match.

— Filter based on feature scores

This filter restricts the list of pairs to a specific score value or a range of values for each feature. The “Empty” and “Non-empty” filter types can be used for each feature to filter the list of pairs for a specific subset (see the [Subsets](#) for more information). First, select the type of the filter. If the Non-empty filter type is selected, you can edit the Min Score and Max Score values to reduce the range of the displayed feature scores.

In addition to using the Filter dialog, you can see and change filters for the Review, Dataset, Current Label, Training Label, and Prediction columns by right-clicking the column title and selecting a filter for that column. You can also sort the list of record pairs by any fixed column using the same column context menu.

Additional filters are available by right-clicking any record pair in the list:

— Show pairs for same subset

This shows a filtered list of pairs that belong to the same subset of present feature scores. This filter is useful to analyze all pairs for a single subset to find similar pairs and any inconsistently labeled pairs. A check mark is displayed at this item of the context menu when this filter is active. You can select the same menu item to clear the filter.

— Show confirming pairs

This shows a filtered list of pairs with feature scores that confirm the label of the selected pair. Such pairs provide evidence to the model that helps correctly classify the selected pair. The selected record pair is always included in the list of confirming pairs. If a pair from the validation dataset has no confirming pairs other than itself, this might be the reason why the model prediction is incorrect. Adding similar training pairs that confirm the selected pair can teach the model to classify the selected pair correctly.

— Show contradicting pairs

This shows a filtered list of pairs with feature scores that contradict the label of the selected pair. For example, if the selected pair is labeled "False", then another pair where all feature scores are lower than the ones in the selected pair should not be labeled True, because the second pair is “even more false”. If any pair has contradicting pairs, this is the likely reason why the model prediction is incorrect. You must review all contradicting

pairs and change the labels of the contradicting pairs or the label of the selected pair appropriately to make all labels consistent.

i Note: The applied filter for the same subset, confirming or contradicting pairs is shown in the Filter dialog as a collection of filters that are based on feature scores and the label. You can review and further modify these filters in the Filter dialog.

Back

By using the Back button you can view the previously used filters such as, list of errors, confirming pairs, and contradicting pairs filters.

i Note: The filter history is not saved in the Learn project, thus opening the same project again does not have access to the history of previous filters.

i Note: The state with no filters such as after clicking the Reset button, is also treated as one of the filters in the filter history.

Reset

The Reset button removes all current filters and sort orders to display the list of all saved pairs. It also removes any automatic filters that might have been applied while processing model training suggestions or by clicking on links in the Training or Trained Model tabs.

Training a Learn Model

This section explains how to start and stop the model training process, save the resulting trained model, and inspect model training results.

Training a Learn Model

Training a Learn model is designed to teach the model to predict the labels of novel record pairs based on the example labeled pairs provided in the Training dataset.

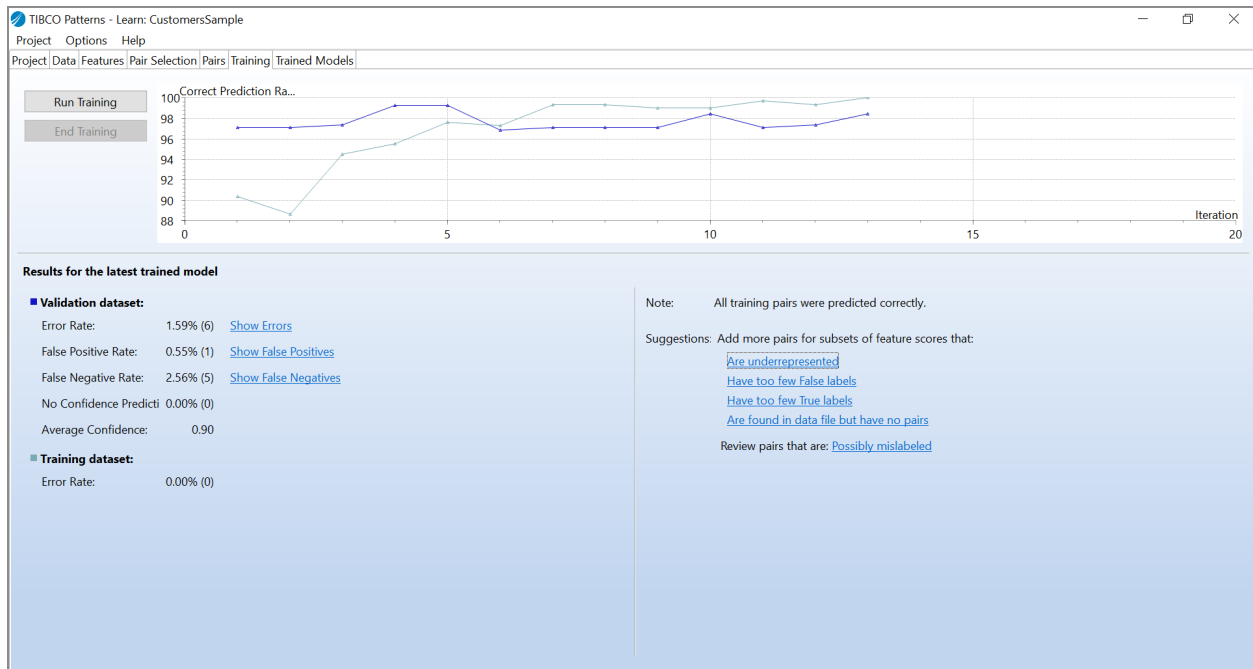
If the predicted label and the assigned label do not match, the trained model generates a classification error. Classification errors are made for a number of reasons: the pair might be mislabeled, there is a human error in labeling similar pairs consistently, or the subset or situation that this pair belongs to is not represented or underrepresented in the training dataset. Examine the pairs with errors and change some labels to make all labels consistent. You can use the filter to display pairs that contradict any misclassified pair on the Pairs tab, see the section [Filtering and sorting record pairs](#). You can also add more similar pairs so that the model learns this situation better.

The model is typically trained in two-passes, the first pass is indicated by thin graph lines and the second pass is indicated by broad graph lines. The goal of training is to minimize the validation error rate, thus the second pass stops at the iteration where the validation error rate is the lowest. If there are several iterations with equal lowest validation error rate, the second pass stops at the iteration where the training error rate is the lowest. The second pass is not used if the last iteration is already the best iteration.

Run Training

This button starts the model training process that usually takes many iterations. Statistics of the model are displayed after each training iteration.

Figure 16: Run Training



End Training

Click the End Training button to terminate model training. If training iterations have started, the training stops after completing the current training iteration.

Note: The correct prediction rate of such model is most likely not optimal, hence stopping the training prematurely is not recommended. However, you can still save this model if you are satisfied with the model training results.

Save Model and Generate Suggestions

Click the **Save Model** button to save the current model and model scores for each record pair. Suggestions to improve the model are generated and displayed. Click the suggestion link to perform the suggested action. After opening the project again, you can click the Generate Suggestions button to display the list of suggestions without retraining the model.

Model Training Results

After the model training ends, one of the following training results is displayed in the Note field:

- Best iteration was found

The iteration with the lowest validation error rate was found. This is the best result.

- All training pairs were predicted correctly

The model was able to correctly predict the labels of all pairs in the training dataset, and the validation error rate at the last iteration is the lowest. The second training pass is not needed. Review any incorrectly classified pairs in the validation dataset and then add more pairs that are similar to the incorrectly classified pairs.

- User ended the training

The model training was terminated prematurely. It is best to not use models, which are not optimal.



Note: The results of the training might be different from the results of training the model in the same Learn project using a previous version of TIBCO Patterns.

Results for Validation Dataset

The following Validation dataset results are displayed during and after the training. Most results display a percentage followed by the number of pairs in brackets:

Error Rate: gives the proportion of incorrectly classified record pairs.

False Positive Rate: gives the proportion of record pairs that are labeled 'False', but are classified as 'True'.

False Negative Rate: gives the proportion of record pairs that are labeled 'True', but are classified as 'False'.

No Confidence Predictions: gives the proportion of record pairs that cannot be reliably predicted by this trained model because they have zero confidence and belong to an untrained or completely contradictory area of the model.

Average Confidence: displays the mean confidence of all model predictions for record pairs. The model reports each predicted score with a certain confidence value between 0.0 (for an untrained or completely contradictory prediction) and 1.0 (for a very confident prediction). The prediction confidence for a pair from a specific subset is higher when more pairs with similar feature scores and non-contradicting labels for that subset are used in training.

Click any link in the Validation dataset section to display the appropriately filtered list of pairs. For more information, see [Reviewing Record Pairs](#).

Figure 17: Errors in Validation Dataset After Clicking Show Errors Link

TIBCO Patterns - Learn: CustomersSample

ProjectOptionsHelp

ProjectDataFeaturesPair SelectionPairsTrainingTrained Models

Errors in Validation Dataset

Displaying 6 record pairs

BackDeleteDelete AllShow/Hide FieldsFilterReset

Review	Data Set	Current Label	Training Label	Model Score	Prediction	Confidence	key	first_name	middle_name	last_name	birth_date	address	city	state	zip	phone
<input type="checkbox"/>	Vld	True	True	0.362	Incorrect	0.055	200206	CLAIRE		MAXINE	1958/06/11		WEST FALLS	NC	70301	0237731789
							200207	CFKLAIRE		MAXINE	1960/07/12	1104 YGUNDERSON DR	WEST FALLS	NC	70301	8023731789
<input type="checkbox"/>	Vld	True	True	0.340	Incorrect	0.170	200214	VALERIE	OCTAVIA	SHERWOOD	1958/10/30		RED LEVELL		21412	
							200215	VALERIE	COCAVIA	SHERWOOD	1959/11/30	1832 W SOCKWELL ST	PFEIFER		21412	
<input type="checkbox"/>	Vld	True	True	0.431	Incorrect	0.187	538	ALBERTO	LANDOON	OROZCO	1968/12/15		GUYS	MI	50567	3167559167
							200230	ALBERTO	LANDOON	OROZCO	1967/01/15		GUYS	MI	50567	3167559167
<input type="checkbox"/>	Vld	False	False	0.550	Incorrect	0.109	24241		BXENNY	BALSTER		1483 SMITH RIDGE RD	RIRONWODOD	MD	32530	7547828581
							206193			MERIGOLD		1483 SMITH RIDGE ZRD	RIRONWODOD	MDD	3253	75478258581
<input type="checkbox"/>	Vld	True	True	0.053	Incorrect	0.101	119664		MURRAY	GVOVEA	1972/01/20	1071 MINERS DR E		NRJ		9498500806
							229888		MRRAY	GVOVEA	1972/03/20	1071 MINERS DR E		RNRJ		9498500806
<input type="checkbox"/>	Vld	True	True	0.309	Incorrect	0.060	212479	MINH		JEAN	1964/06/29	1315 S 850 E	JENKINQJONES	PA	54124	51598975314
							212480	JEAN	MINH			J1315 S 850 E	JENKINQJONES	PA	54124	51959897534

Feature Scores for selected Record Pair

Name: First Name	Name: Last Name	Name: Cognate Name	Address	City	Phone	Zip	dob
0.8311196	0.032929	0.672336	-1.000000	1.000000	0.808710	1.000000	0.250000

Results for Training Dataset

The following result for the Training dataset is displayed during and after the training:

Error Rate: gives the proportion of incorrectly classified record pairs.

Clicking the Review Labels link provided in this section displays the incorrectly predicted pairs in the Training dataset. Refer to [Reviewing Record Pairs](#) for more information. If there are errors made in the Training dataset, you should always review the labels of these pairs to make sure they are correct and consistent.

Model and Training Options

This section explains model and training options in the Learn UI.

Small Learn Models

A small Learn model exhibits only a simple exponential growth as the number of features increases. The size of the model grows slower with each added model feature. This results in much smaller model files for the same number of features, and thus allows more features to be supported, given the same limitations on the available memory. The accuracy of a trained small Learn model is comparable with the accuracy of the large model type for most datasets, although in rare cases a large model might classify some situations better.

To reduce model training time, dynamic subset training is automatically switched off starting at 17 features. Therefore, a 17-feature model is expected to train much faster than a 16-feature model.

Adaptive Parameters for Model Creation

Model creation parameters are automatically adjusted by default based on the number of model features. This adjustment does not allow the size of the model to exceed 2.8 GB for all models that have up to 18 model features.

Model creation parameters are not adjusted automatically if the user selected manual options in the Model Options dialog and clicked the OK.

i Note: Using 14 or more model features can result in long model-training times (one hour or more in some cases). Using more than 18 model features can result in extremely large model files and out-of-memory errors.

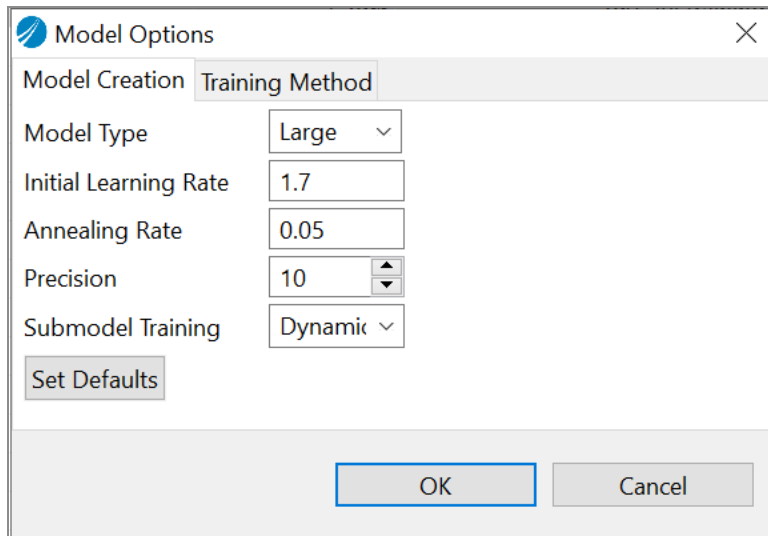
Setting the Model and Training Options

You can manually set the model creation and training options for your project in Learn UI.

To set the model creation and training options in Learn UI:

1. On the top menu, select Options >Model item.

Figure 18: Model Options



2. On the Model Creation tab, adjust the following options that are used to create the model:

Model Creation Options

Field	Description
Model Type	<p>Large - Large models take subsets into account; they might predict better in some situations.</p> <p>Small - Small models are much smaller and can support more features.</p>
Initial Learning Rate	<p>Defines how quickly the model state changes during training. It is recommended to set the value between 1.1 and 2.0.</p> <p>Default value:</p> <ul style="list-style-type: none"> — For large Learn Model type - 1.7 — For small Learn Model type - 1.5

Field	Description
Annealing Rate	Defines the speed of learning state decrease with each iteration. The default value of 0.05 makes the learning rate two times smaller after about 15 iterations.
Precision	Precision of internal model weights. A higher precision might help distinguish similar examples with different labels. It is recommended to set the value between 6 and 10 (default). The amount of memory used by the model is proportional to 2 to the power of the selected precision value.
Submodel Training	<p>Dynamic - (Default) This mode uses augmentation of training data to better train submodels. It is recommended for most projects, especially when training with relatively few record pairs.</p> <p>None - This option trains with actual training examples only. It can be used if datasets contain abundant examples from all subsets that can be encountered. Training is much faster; therefore, this mode is also applicable when the number of model features is very large.</p>



Warning: Using unreasonable combinations of model creation options combined with a large number of model features can result in long model training times or extremely large model files and out of memory errors.

3. Optionally, click the Set Defaults to set the default model creation options that are based on the current number of model features.
4. To set the desired model training options, click the Training Method tab.

Figure 19: Training Method Options

The screenshot shows a dialog box titled "Model Options" with a close button (X) in the top right corner. It has two tabs: "Model Creation" and "Training Method", with the latter being the active tab. Inside the "Training Method" tab, there are two radio buttons: "Minimize validation error" (which is selected) and "Minimize training error (do not use for final model)". Below these are three input fields: "Iterations to explore after finding best iteration" with a value of 35, "Minimum number of iterations" with a value of 0, and "Good fit distance between error rates" with a value of 1.0 and a percentage sign (%). At the bottom left is a "Set Defaults" button, and at the bottom right are "OK" and "Cancel" buttons.

5. On the Training Method tab, select Minimize validation error to use the recommended method that avoids overfitting. The training stops at the iteration where the validation error is the lowest (other criteria for tie breaking are also used). Use this option for the final model that is going to be used in production.
6. Select Minimize training error to stop the training at the iteration where the training error rate is the lowest. You can use this method to identify mislabeled pairs in the training dataset. The remaining errors in the training dataset are the minimal set of pairs that the model was not able to predict correctly, thus these pairs are likely to have incorrect or contradictory labels.

It is recommended not to use this option for the final model, or for evaluating the performance of the model, because of significant overfitting that might happen in such training.

7. Adjust the following information in the fields:

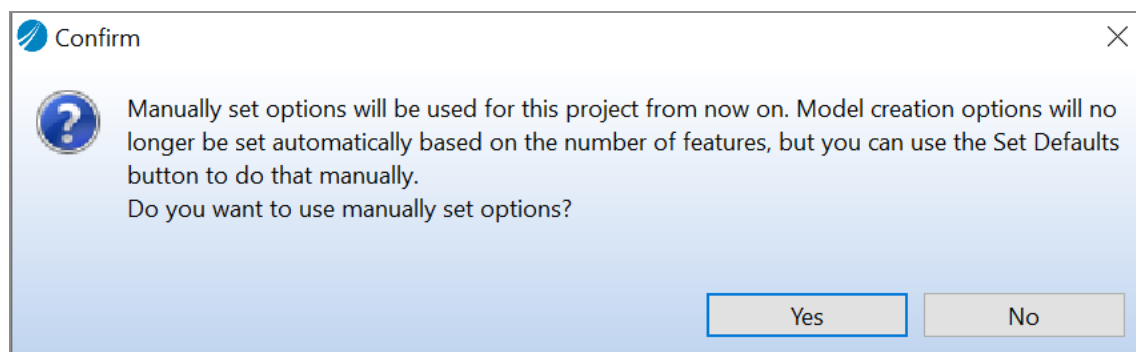
Figure 20: Training options

Field	Description
Iterations to explore after finding best iteration	Perform this number of iterations after the current best result is found to search for a better result. The default is 35 iterations. The value can be increased to explore more iterations, especially if the learning rate is small. Using a much smaller value is not recommended.

Field	Description
Minimum number of iterations	The minimum number of iterations that are always performed when training the model. This can be used to extend training until the learning rate becomes reasonably small. The default is 0 iterations. Changing this parameter is rarely needed because typically the training is made long enough by using the recommended value of the Good fit distance between error rates parameter.
Good fit distance between error rates	Training does not stop at iterations where training error rate is greater than the validation error rate by more than the specified distance. The default value is 1%. Essentially, if 0% is used, the training continues until the model is able to predict the training dataset as well as, or better than, the validation dataset. This parameter should be 0% or slightly above 0% to prevent underfitting. Using 100% ignores underfitting (not recommended).

8. Optionally, click the Set Defaults to set the default training method options.
9. Click OK. If you have not used manually set model options for this project before a confirmation dialog is displayed and the manually set options are used from that point on in this project. In this case model creation options are no longer automatically adjusted based on the number of features, even if the number of features is later changed. If you intend to keep using the automatic adjustment of model creation options and are using the dialog just to view the default options for the current number of features, click Cancel instead.

Figure 21: Confirmation



All the model setting and training method options are saved when the project itself is saved.

Processing Training Suggestions

The training suggestions help you determine whether your model is well-trained. They help you improve the accuracy of the trained model in a variety of situations.

Based on the results of model training, suggestions are provided to improve the model by adding specific types of training pairs or reviewing the labels of specific existing training pairs that appear to be inconsistent.

Determining if a Model is Well Trained

When the Learn UI can no longer generate more specific suggestions, the Suggestions section displays that the existing pairs provide sufficient coverage. In addition, observe the reported training result of the model in the Note section on the Training tab. The training result Best iteration was found is the desired result, which means that the iteration with the lowest validation error rate was found.

One of the best ways to ensure that the model is well trained is to use the Low Confidence Pair Finder on the Pair Selection tab with a table of sufficient size until it can no longer find any new low confidence pairs to be labeled. This can address untrained and undertrained situations, even if such situations are not currently represented in Training and Validation datasets. For more information, see the section [Finding Useful Pairs Automatically](#).

For the model to be well trained, the Training and Validation datasets should not have any pairs that contradict one another. See [Reviewing Contradictory Pairs](#) section for the method to resolve such contradictions.

When the model is well trained, the current trained model can be exported and applied for solving real-world record matching problems. See [Exporting a Model](#) section for details. You can also export the model when a training result other than "Best iteration was found" is reported if you are satisfied with the model performance statistics.

Handling Suggestions

The Learn UI displays suggestions to guide the next actions that you should perform to improve the performance of the trained model.

Model training suggestions are provided on the Training tab. The Learn UI shows suggestions related to the latest trained model. A suggestion can request you to add more pairs to specific subsets, or review the labels of some existing pairs.

Retraining a Model

After adding more pairs, or changing pair labels, click the Run Training button and the model is retrained. The current saved model is overwritten when the new model is saved. A new set of suggestions is generated.

Types of Training Suggestions

The training suggestions are calculated by analyzing the performance of the current trained model as well as the currently defined features and record pairs. Several types of suggestions are provided.

Types of Training Suggestions

Suggestion Group	Types	Description
Suggestions to Add Pairs to Specific Subsets	Adding Pairs to Subsets that Have Validation Pairs but No Training Pairs	Used for an untrained subset. It is a subset, which has not been trained with any training examples, because one or more record pairs for that subset are present only in the validation dataset.
	Adding Pairs to Underrepresented Subsets	Used for underrepresented subsets. These are determined by the relatively small number of record pairs for those subsets in the training dataset.
	Adding Pairs to Subsets that Have Too Few True/False Labels	Used for subsets where the percentage of the less frequent label is below a certain threshold. It is important to maintain a balance between "True" and "False" labeled pairs.

Suggestion Group	Types	Description
	Adding Pairs to Subsets that are Found in Data File but Have No Pairs	Used for subsets that have no pairs in either dataset. These subsets are found by analyzing all records in the data table.
Suggestions to Review Existing Record Pairs	Reviewing Possibly Misabeled Pairs	Used in scenarios where record pairs are presumably mislabeled. The labels of such pairs are different from most labels of similar pairs in the same subset. Such pairs are presented for user review.
	Reviewing Contradictory Pairs	Used to review record pairs that have other pairs that contradict them. Contradictory pairs should be avoided for optimal training results.

Suggestions to Add Pairs to Specific Subsets

These suggestions are generated by analyzing the existing record pairs for each subset. The user is asked to add more pairs to specific subsets.

The suggestions in this group are:

Adding Pairs to Subsets that Have Validation Pairs but No Training Pairs

An untrained subset is a subset that has not been trained with any training examples, because one or more record pairs for that subset are present only in the Validation dataset.

Some subsets that do not have any pairs in the training dataset might still be reasonably trained. This is because the model is being trained with some automatically generated examples that belong to these subsets. Nevertheless, it is recommended to have some training record pairs for all subsets that are present in the Validation dataset.

Adding Pairs to Underrepresented Subsets

These subsets are determined by the number of record pairs for each subset in the training dataset. The subsets that have very few record pairs also tend to have low prediction confidence values (but confidence depends on other factors as well).

Underrepresented subsets are those that have significantly fewer pairs than other subsets in the training dataset. The training process can cause underrepresented subsets to be influenced too much by other subsets that have more pairs. Therefore, model predictions for the underrepresented subsets can be improved by adding more training record pairs to these subsets.

Adding Pairs to Subsets that Have Too Few True/False Labels

These are subsets in the training dataset that have a large imbalance between “True” and “False” labeled pairs. To train a model properly, there should be at least some balance between “True” and “False” labeled pairs for each subset. You can either add more pairs with the underrepresented label or, if necessary, delete some pairs with the overrepresented label.

If you are sure that no pair from a certain subset can ever be labeled as "True", it is recommended to mark a pair that belongs to such subset as "Always False" on the Pair Selection tab. For more information, see the section [Always False Subsets](#).

If you use the Low Confidence Pair Finder to automatically find the majority of pairs, note that it tends to find most of the pairs that are assigned a False label. In this case this suggestion can largely be ignored.

Adding Pairs to Subsets that are Found in Data File but Have No Pairs

These are subsets that exist in the data table but have no training and no validation pairs. The model cannot reliably predict matches for subsets for which it has not been trained. You likely require that pairs with all types of records that exist in your data table be reliably predicted by the trained model. Therefore, it is recommended that pairs for all subsets that exist in the data table are added to the training dataset.

Processing a Suggestion to Add Pairs

Each suggestion is displayed as a link on the Training tab. When you click this link, the Learn UI shows the Pair Selection tab which is specially configured for processing that suggestion:

1. The data table is filtered to display records for the first subset in the family of subsets for this suggestion.
2. The caption of the table indicates the current subset. For example, “Add more pairs to subset 0101111 (2 of 15)”. The binary code indicates the present (1) and empty (0) feature scores in this subset.
3. The Previous and Next buttons are displayed above the data table, if the suggestion has more than one subset. You can use these buttons to filter the data table for each individual subset in the family of subsets for the current suggestion.
4. Use the Reset button to go back to the unfiltered data table and exit the suggestion processing mode.

You should add a few training record pairs to the current subset. Click Next to see the filtered records for the next subset, and so on until all the subsets in the selected suggestion have been processed. A warning is displayed while proceeding to the next subset if no record pairs were added to the training dataset for the current subset.

Some subsets might have zero or only one record in the table, which is not enough to create a record pair. An appropriate message is displayed, and such subsets can be skipped. You might be able to manually find records in the table that have different empty and non-empty fields. Then you can still create a record pair from the given subset by combining such records.

i Note: New pairs are added randomly to either the Training dataset or the Validation dataset. It is the Training dataset that must actually change to see different model training results for that subset. After saving a record pair, the Pair Selection tab indicates the dataset the last pair has been added to.

Filtering Record Pairs for the Suggestion Subset

You can view the filtered list of existing pairs for a specific subset while processing one of the Add more pairs suggestions for that subset. Click the Add more pairs to subset <subset code> link displayed in the Pair Selection tab. This opens the Pairs tab displaying the existing record pairs in the subset. After reviewing the pairs, return to the Pair Selection tab to continue processing the suggestion.

Suggestions to Review Existing Record Pairs

These suggestions identify existing pairs that need to be reviewed by the user.

Reviewing Possibly Mislabeled Pairs

This suggestion is for reviewing potentially mislabeled pairs. It is based on the distance (absolute difference) between the actual label and the score provided by the trained model. The actual label represents score 0.0 or 1.0 for “False” and “True” labels respectively. Record pairs where the desired score for the actual label and the predicted score are very distant are included in the suggestion.

Reviewing Contradictory Pairs

This suggestion is for reviewing pairs that have other pairs that contradict them. Based on the feature scores, the labels of such pairs contradict one another. For optimal model performance it is essential to resolve such contradictory pairs by changing the labels of certain pairs. If this suggestion is not provided, the datasets that were used in training the model contain no contradictory pairs.

When reviewing the list of contradictory pairs, you should analyze each pair individually. Right-click a pair and select [Show contradicting pairs](#) to see only the pairs that contradict the selected pair (see the section [Filtering and sorting record pairs](#)). After examining these pairs, you can make the labels consistent by either changing the label of the original pair, or changing the labels of all pairs that contradict the original pair. Use the Back button to go back to the overall list of contradictory pairs. Pairs are not immediately removed from the list of contradictory pairs after changing their labels. Therefore, after changing several labels, retrain and save the model, then inspect any remaining contradictory pairs.

Processing a Suggestion to Review Pairs

Click the suggestion link to display the Pairs tab with the list of pairs that contains only the record pairs included in that suggestion. You should review and change labels where necessary. The Reset button in the Pairs tab removes the suggestion filter and displays the list of all record pairs.

Working with Trained Models

This section discusses reviewing statistics for saved trained models, adjusting the cutoff score, testing, and exporting a model.

Trained Model

The Trained Model tab displays statistics for all trained models that were saved. The first item in the list always represents the current model. Every time a model is trained and saved, the current trained model (if it exists) is overwritten with the new model. Many functions on this tab are enabled only for the current model.

Figure 22: Trained Model Tab

The screenshot shows the TIBCO Patterns - Learn: CustomersSample application. The 'Trained Models' tab is active, displaying a table of saved models. Below the table, there are sections for 'Display results for dataset:', 'Testing Results', and 'Query Cutoff Score'.

Model Name	Training Date	Error Rate	Training Pairs	Notes
current Save a copy	5/29/24 11:43 AM	1.59%	290	All training pairs were predicted correctly.
CustomerModel1	5/29/24 11:43 AM	1.59%	290	All training pairs were predicted correctly.

Display results for dataset:

- ☒ Saved validation dataset 378 pairs
- ☐ Saved training dataset 290 pairs
- ☐ Latest manual test using:

Testing Results

Error Rate:	1.59%	Show Errors
False Positive Rate:	0.55%	Show False Positives
False Negative Rate:	2.56%	Show False Negatives
No Confidence Predict:	0.00%	
Average Confidence:	0.90	

Query Cutoff Score

Cutoff Score: [Optimize](#) [Default](#)

Adjusted rates for validation dataset:

Error Rate:	1.59%
False Positive Rate:	0.55%
False Negative Rate:	2.56%

[Export Executable Model](#)

On this tab you can perform the following activities:

- Save a copy of the current trained model. See [Saving and Reviewing Trained Models](#) for more information.
- Test the model using a selected set of record pairs. See [Testing a Trained Model](#) for more information.

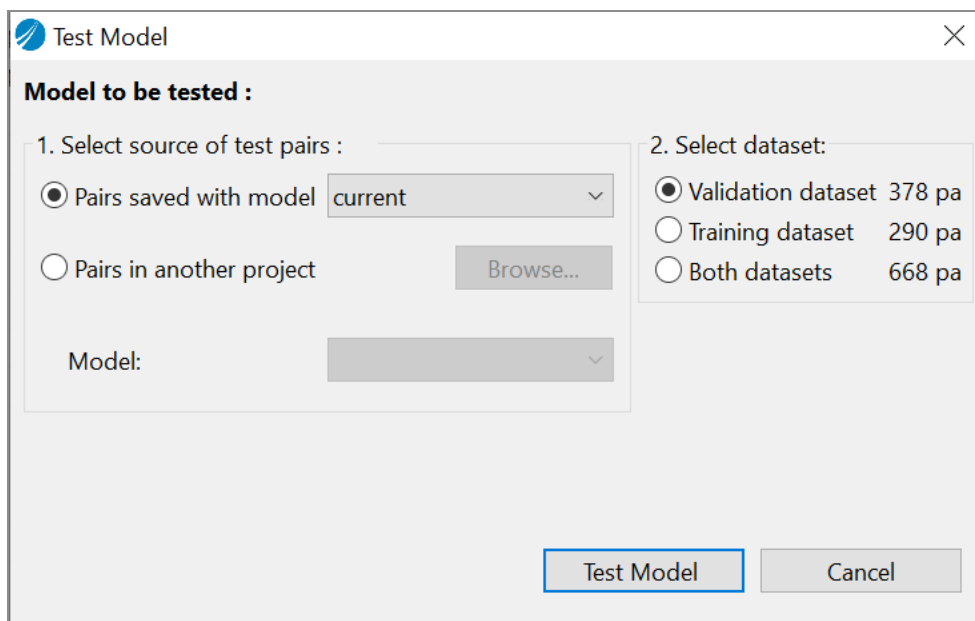
- Set query cutoff score to adjust the balance of false positives to false negatives. Depending on the application, it can be better to reduce false positives at the cost of increasing false negatives or vice versa. See [Setting the Query Cutoff Score](#) for more information.
- Export a model to be used for the prediction of different record pairs from real-world data. See [Exporting a Model](#) for more information.

Testing a Trained Model

The performance of the current trained model can be tested using various sets of record pairs. Select the current model and then click Run a test... to open the dialog for selecting the dataset of record pairs to be used for testing.

First, select the source of the record pairs used for testing. The source can be one of the models saved in the current project. It can also be a model saved in a different project, provided that the other project uses the same data file as the current project. Next, you can select a specific dataset or all pairs from both datasets.

Figure 23: Testing a Trained Model



Test Model

Click this button to load the trained model, use that model to predict all labels for the record pairs in the selected datasets, and display the results on the Trained Model tab. Results of this test are shown only when you select the Latest manual test radio button.

Setting the Query Cutoff Score

This functionality is used to find the optimum model score cutoff (the score that separates the two labels) for your application. It does not change the constant threshold score of 0.5 that is used to separate the two labels during model training. The Query Cutoff Score group box shows the adjusted error rates for the validation dataset if only record pairs with the prediction score at or above the cutoff score were classified as matches. The adjusted false positive rate shows the percentage of predicted matches that are not true matches. The adjusted false negative rate shows the percentage of true matches that would be classified as false. The adjusted error rate shows the overall percentage of incorrect results for the current cutoff score.

In most situations it is a best practice that you start using the model with the default cutoff score of 0.5, which reflects actual predictions made by the model and usually minimizes the overall error rate. However, your particular application might require that either false positives or false negative be reduced to an absolute minimum. Using this function you can determine the suitable cutoff score that achieves the required false positive or false negative rate.

The Optimize button sets the cutoff score to minimize the overall error rate. The Default button sets the cutoff score to the training threshold level of 0.5.

Figure 24: Query Cutoff Score

Query Cutoff Score

Cutoff Score

Adjusted rates for validation dataset:

Error Rate:	7.14%
False Positive Rate:	0.00%
False Negative Rate:	25.00%

Saving and Reviewing Trained Models

The Trained Model tab displays a list of saved models. You can select a model and review its results for training and validation datasets.

Saving a Copy of Current Model

Click the Save a copy link to save a copy of the current trained model. This also saves a copy of all field types, features and record pairs associated with the current model. You can later compare the statistics of the saved model with a newly trained current model. See [Trained Model Tab](#) figure for an example of multiple saved models.

Reviewing Results of a Trained Model

The Trained model tab displays the results of any trained model that was saved.

Figure 25: Review Results

Display results for dataset:	Testing Results		
<input checked="" type="radio"/> Saved validation dataset 378 pairs	Error Rate:	0.53%	Show Errors
<input type="radio"/> Saved training dataset 290 pairs	False Positive Rate:	0.55%	Show False Positives
<input type="radio"/> Latest manual test using:	False Negative Rate:	0.51%	Show False Negatives
	No Confidence Predictions:	0.00%	
	Average Confidence:	0.89	

Displaying Results for a Specific Dataset

After selecting a trained model in the list, you can display results for one of the following:

- The validation dataset saved with the selected model.
- The training dataset saved with the selected model
- The latest manual test that used a specific dataset from a selected project and model. See [Testing a Trained Model](#) for details.

Testing Results

This section displays statistics of model predictions for the selected dataset:

- Error Rate
- False Positive Rate
- False Negative Rate
- Percentage of No Confidence Predictions
- Average Confidence

Refer to [Results for Validation Dataset](#), for more information.

Links provided can be used to review a filtered list of pairs that correspond to the specific item in the “Testing Results” section.

Deleting a Model

To delete a saved model together with features and datasets associated with it, do the following:

1. Select the saved model in the grid.
2. Click Delete.
3. Click Yes to confirm your decision.

The selected saved model and datasets related to it are deleted.



Note: If the current model is selected, this button only deletes the trained model and model results, but features and datasets are preserved. Typically this significantly reduces the size of the project directory.

Exporting a Model

This function exports all files that are needed to deploy a trained model to a TIBCO Patterns server. The following files are exported for the selected model:

- The trained model binary file
- A Java source file that contains the TIBCO Patterns query with all model features
- All thesaurus files used by the TIBCO® Patterns Search query

Figure 26: Exporting a Model

Export

Export Executable Model

Export binary model, query and thesaurus files

Model File Name:

Query Class Name:

Location:

To export a model do the following:

1. Select the model to be exported.
2. Click Export Executable Model.
3. Specify the names of the model and query files and the location for the exported files.

i Note: Make sure that the specified directory does not contain any thesaurus files that should be preserved because any thesaurus files with the same names will be overwritten.

4. Click Export.

The exported model can then be loaded to a TIBCO Patterns server. You should also load a table with fields that are identical to the ones in the data table used to train the model. In addition, all thesaurus files used by the query should be loaded to the server. The thesaurus name on the server must be the same as the thesaurus file name.

In a typical scenario, you use the exported Java method to obtain a `NetricsQuery` object with values from one record of the loaded table, or with values of a new incoming record. Then you can perform a search operation on the server table using that query, which in turn uses the Learn model that you have loaded to the server to predict the final query result. Thus the query finds records in the table that are predicted by the model to be the most similar to the record used to create the query. This allows you to use the model for finding likely duplicate records in the table. For more information on the relevant operations with the server, see TIBCO Patterns Concepts Guide.



Note: The Query file name is also used as the name of a Java factory class. This class provides a method that generates a NetricsQuery object, representing the query used to match a record with the given set of field values.

TIBCO Documentation and Support Services

For information about this product, you can read the documentation, contact TIBCO Support, and join TIBCO Community.

How to Access TIBCO Documentation

Documentation for TIBCO products is available on the [Product Documentation website](#), mainly in HTML and PDF formats.

The [Product Documentation website](#) is updated frequently and is more current than any other documentation included with the product.

Product-Specific Documentation

Documentation for TIBCO® Patterns is available on the [TIBCO® Patterns Product Documentation](#) page.

How to Contact Support for TIBCO Products

You can contact the Support team in the following ways:

- To access the Support Knowledge Base and getting personalized content about products you are interested in, visit our [product Support website](#).
- To create a Support case, you must have a valid maintenance or support contract with a Cloud Software Group entity. You also need a username and password to log in to the [product Support website](#). If you do not have a username, you can request one by clicking **Register** on the website.

How to Join TIBCO Community

TIBCO Community is the official channel for TIBCO customers, partners, and employee subject matter experts to share and access their collective experience. TIBCO Community offers access to Q&A forums, product wikis, and best practices. It also offers access to extensions, adapters, solution accelerators, and tools that extend and enable customers to gain full value from TIBCO products. In addition, users can submit and vote on feature requests from within the [TIBCO Ideas Portal](#). For a free registration, go to [TIBCO Community](#).

Legal and Third-Party Notices

SOME CLOUD SOFTWARE GROUP, INC. (“CLOUD SG”) SOFTWARE AND CLOUD SERVICES EMBED, BUNDLE, OR OTHERWISE INCLUDE OTHER SOFTWARE, INCLUDING OTHER CLOUD SG SOFTWARE (COLLECTIVELY, “INCLUDED SOFTWARE”). USE OF INCLUDED SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED CLOUD SG SOFTWARE AND/OR CLOUD SERVICES. THE INCLUDED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER CLOUD SG SOFTWARE AND/OR CLOUD SERVICES OR FOR ANY OTHER PURPOSE.

USE OF CLOUD SG SOFTWARE AND CLOUD SERVICES IS SUBJECT TO THE TERMS AND CONDITIONS OF AN AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER AGREEMENT WHICH IS DISPLAYED WHEN ACCESSING, DOWNLOADING, OR INSTALLING THE SOFTWARE OR CLOUD SERVICES (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH LICENSE AGREEMENT OR CLICKWRAP END USER AGREEMENT, THE LICENSE(S) LOCATED IN THE “LICENSE” FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE SAME TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

This document is subject to U.S. and international copyright laws and treaties. No part of this document may be reproduced in any form without the written authorization of Cloud Software Group, Inc.

TIBCO, the TIBCO logo, the TIBCO O logo, ActiveMatrix BusinessWorks, BusinessConnect, TIBCO Hawk, TIBCO Rendezvous, TIBCO Administrator, TIBCO BusinessEvents, TIBCO Designer, and TIBCO Runtime Agent are either registered trademarks or trademarks of Cloud Software Group, Inc. in the United States and/or other countries.

All other product and company names and marks mentioned in this document are the property of their respective owners and are mentioned for identification purposes only. You acknowledge that all rights to these third party marks are the exclusive property of their respective owners. Please refer to Cloud SG’s Third Party Trademark Notices (<https://www.cloud.com/legal>) for more information.

This document includes fonts that are licensed under the SIL Open Font License, Version 1.1, which is available at: <https://scripts.sil.org/OFL>

Copyright (c) Paul D. Hunt, with Reserved Font Name Source Sans Pro and Source Code Pro.

Cloud SG software may be available on multiple operating systems. However, not all operating system platforms for a specific software version are released at the same time. See the “readme” file for the availability of a specific version of Cloud SG software on a specific operating system platform.

THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT.

THIS DOCUMENT COULD INCLUDE TECHNICAL INACCURACIES OR TYPOGRAPHICAL ERRORS. CHANGES ARE PERIODICALLY ADDED TO THE INFORMATION HEREIN; THESE CHANGES WILL BE INCORPORATED IN NEW EDITIONS OF THIS DOCUMENT. CLOUD SG MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE PRODUCT (S), THE PROGRAM(S), AND/OR THE SERVICES DESCRIBED IN THIS DOCUMENT AT ANY TIME WITHOUT NOTICE.

THE CONTENTS OF THIS DOCUMENT MAY BE MODIFIED AND/OR QUALIFIED, DIRECTLY OR INDIRECTLY, BY OTHER DOCUMENTATION WHICH ACCOMPANIES THIS SOFTWARE, INCLUDING BUT NOT LIMITED TO ANY RELEASE NOTES AND "README" FILES.

This and other products of Cloud SG may be covered by registered patents. For details, please refer to the Virtual Patent Marking document located at <https://www.cloud.com/legal>.

Copyright © 2010-2024. Cloud Software Group, Inc. All Rights Reserved.