



---

# User Manual

*Version 1.0*

## Blueocean Market Intelligence



About UTAP .....	3
User Manual.....	3
1. CREATING MODELS .....	3
Data Extraction.....	3
Data Sampling .....	3
<i>Predictive Variable:</i> .....	4
<i>Dependent Variable:</i> .....	4
<i>Training set</i> .....	4
Data cleaning .....	4
<i>Stop words</i> .....	4
<i>Junk words</i> .....	4
<i>Lemmatization</i> .....	4
Training .....	5
<i>TF-IDF</i> .....	5
<i>Naive Bayes</i> .....	5
<i>Support Vector Machine</i> .....	5
<i>Stochastic gradient descent</i> .....	5
<i>Logistic Regression</i> .....	5
<i>Ridge Classifier</i> .....	5
Validation .....	6
Frequently Asked Questions .....	7



## About UTAP

UTAP is a web based platform which helps you to categorize, and structure your unstructured textual data to aid generation of actionable insights to aide further analysis. You can play with your text the way you want to. UTAP provides pre-designed and pre-defined options which let you customise the standard steps in a typical text classification exercise.

## User Manual


This document works with the purpose of familiarizing users with the various modules, and components of UTAP, the web based text analytics platform developed by Blueocean Market Intelligence™.

### 1. CREATING MODELS

This module is designed to help the user build models out of your text.

#### Data Extraction

Data extraction in UTAP has two subsequent steps:

- 
- (a) Entering user credentials
  - (b) Choosing the data and number of records to extract

You can extract data either from a file already present on your computer or from a database by providing user credentials. While extracting from data base you will have to mention the data name and various fields you want to extract.

In addition to choosing the source, you can also mention the number of records to be extracted. You can also mention the number of records to be fetched in the condition.

Preview data on the right side of your screen shows 200 records of your extracted data at a time. The scroll tab lets you view subsequent rows in the data table, and the 'next' option lets you view the next 2000 records.

The 'arrow' acts as a toggle between the two sub-steps of Data extraction.

#### Data Sampling

The user can sample the data between predictor and dependent variable by simple dragging and dropping the data from fields to respective variable.

**Predictive Variable:** It is an independent variable, sometimes also called as experimental variable, is a variable that is being manipulated in an experiment in order to observe the effect on a dependent variable, sometimes called an outcome variable. This variable is often denoted by x.

**Dependent Variable:** This is a variable (often denoted by y ) whose value depends on that of predictor variable.

All the columns in respective variables are supposed to be joined as a single string. *CONCAT* function in the data sampling page helps the user do that.

\*Make sure your field is empty and you have dragged all the data from it.

Whenever the user is building a model, it is a required to split your data into **training and testing**.

**Training set** trains your model and builds it while testing set validates your model. By general standards, most of the data is in the Training set, as the model has to learn from it and remaining data forms the testing set. **Test data** provides you the accuracy of the model.

After concatenating the variables the user is required to split the data into Training & Testing. The slider on the screen helps you divide your data between training and testing. You can also manually adjust it by typing it with the help of edit button.

The preview screen on the right side of the page allows the user to preview the X\_Train, Y-Train, X\_Test, Y\_Test, based on the data split. Preview data on the right side of your screen shows 200 records of your extracted data at a time. The scroll tab lets you view subsequent rows in the data table, and the 'next' option lets you view the next 2000 records.

The 'arrow' acts as a toggle between the two sub-steps of Data sampling.

## Data cleaning

Before any data can be used for model training, it is essential to clean it, to remove junk values, stop words and reduce the words to its root form, to make analysis easier and more relevant.

There are some patterns which are to be omitted and to be replace for which we use **Regular Expression**.

**Stop words** are those who don't have any significance and are to be removed.

**Junk words** are words which do not have significance according to the output required.

**Lemmatization:** It is a predefined function, which allows you to reduce different forms of a word to its root form.

One can remove stop words by inserting a .csv file (containing a list of stop words) in the tab or by manually typing each word. Same is the case with junk words. Similarly, in Regular Expressions the platform enables the user to write the pattern and its replacement manually or provide the same in a .csv file.

The cleaning functions are active only when enabled by clicking on the enabled button. You can disable any of the above cleaning functions by clicking on the disabled button as well. Upon clicking, the clean button will enable the cleaning options chosen.

Choose next to proceed to the training step.

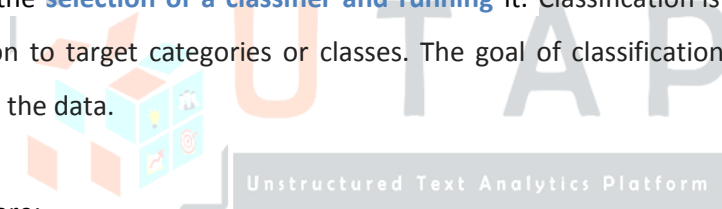
## Training

Training has two steps namely

- (a) Creation of TF-IDF matrix
- (b) Selection of classifier.

**TF-IDF**, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in information retrieval and text mining.

Second step in training is the **selection of a classifier and running** it. Classification is a data mining function that assigns items in a collection to target categories or classes. The goal of classification is to accurately predict the target class for each case in the data.



Various kinds of Classifiers are:

**Naive Bayes:** A formula that calculates a probability by counting the frequency of values and combinations of values in the historical data.

**Support Vector Machine:** Support Vector Machine (SVM) is a powerful, state-of-the-art algorithm based on linear and nonlinear regression. Oracle Data Mining implements SVM for binary and multiclass classification.

**Stochastic gradient descent:** It is a gradient descent optimization method for minimizing an objective function that is written as a sum of differentiable functions.

**Logistic Regression:** A type of generalized linear model that uses statistical analysis to predict an event based on known factors. Also called logistic model/logit model.

**Ridge Classifier:** Ridge Regression is a technique for analysing multiple regression data that suffer from multi collinearity.

For Training, the user is required to first select the directory where he/she wants the models saved.

Next choose the TF-IDF creation step to create the TF-IDF matrix. The parameters are already visible while using this option. You can select or deselect any of these, or use the edit button to enter a new parameter.

The classifier can be run only if the TF-IDF has been created, until which the run button is disabled. After creating the TF-IDF, Run button gets enabled, and any of the available classifiers on which you want your model trained can be selected.

The parameters for all the classifiers are set by default and if the user wants to change the default values you can write in the tab. Now press the run button to create the model. You can check all the scores for training data in the select accuracy matrix step. After choosing any one option in the select accuracy matrix step we can get to the next step by pressing next button.

## Validation

This step is designed to check how accurately your model works. In this step the user can either use the previous data that is already in the system or database for X\_Test and Y\_Test, or can use the existing data option to use the data from the recently sampled data.

Now press the run button to get the Accuracy Score for the train data set. Additional reports such as F1 score, Confusion Matrix, and Classification Report can also be generated in the validation screen as per user requirements.

**F1 score:** The  $F_1$  score (also F-score or F-measure) is a measure of a test's accuracy. It considers both the 'precision  $p$ ' and the 'recall  $r$ ' of the test to compute the score: ' $p$ ' is the number of correct positive results divided by the number of all positive results, and ' $r$ ' is the number of correct positive results divided by the number of positive results that should have been returned. The  $F_1$  score can be interpreted as a weighted average of the precision and recall, where an  $F_1$  score reaches its best value at 1 and worst at 0.

The traditional F-measure or balanced F-score ( $F_1$  score) is the harmonic mean of precision and recall:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

In pattern recognition and information retrieval with binary classification, **precision** (also called positive predictive value) is the fraction of retrieved instances that are relevant, while **recall** (also known as sensitivity) is the fraction of relevant instances that are retrieved.

**Confusion matrix:** Also known as a contingency table or an error matrix, allows visualization of the performance of an algorithm, typically a supervised learning one. Each column of the matrix represents the instances in a predicted class while each row represents the instances in an actual class (or vice-versa). The name stems from the fact that it makes it easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another).

	n=165	
	Predicted: NO	Predicted: YES
Actual: NO	50	10
Actual: YES	5	100

The matrix above shows the actual class v/s predicted class.

**Classification report:** This report displays Precision, Recall, F1-score and support for the classified data. 'Support' mentioned in the report refers to the number of samples of true response that lie in that class.

**Accuracy score:** It is the measure of truly classified classes over total classified classes. From the above confusion matrix it will be:

$$\text{Accuracy} = (50 + 100)/165$$

## Frequently Asked Questions

### 1. What type of file formats are supported by UTAP?

UTAP supports .sql/.csv/.txt/.xls file formats

### 2. What browsers support UTAP?

UTAP is compatible with:

- (a) Mozilla Firefox (Version 41.0.1)
- (b) Google Chrome (Version 46.0.2490.71 m (64-bit))
- (c) Internet Explorer (Version 11)

### 3. Which OS's support UTAP?

UTAP is compatible with the above mentioned browsers on Windows 8.1 and Windows 7

### 4. How does UTAP access data?

Data access will be according to the agreed data sharing and privacy guidelines according to in the scope prior to initiation of the product.

### 5. What is the volume of data that can be handled by UTAP?

There is no upper limit to the volume of data that can be handled by UTAP. However, the taken to extract, pre-process, and train the data may increase according to increase in data volumes.

**6. Is there a golden rule in determining the volume of test data and training data?**

Yes. Generally the data is split into 80-20 i.e. 80% data in training, and 20% in validation/testing or 75-25 i.e. 75% data in training and 25% data in validation/testing.

**7. What insights are provided by UTAP?**

UTAP can provide a classification report which will have each comment/document classified to its designated class/category. Upon detailed analysis categories can be used to predict trends, and progression of categories over time.

**8. How do I measure accuracy in UTAP?**

Accuracy in UTAP can be obtained by clicking on the accuracy score option. Options to download and view the report are also available.

