

ECML PKDD 15 Taxi Trajectory Prediction I Documentation

Generated by Doxygen 1.8.10

Mon Aug 24 2015 11:53:04

Contents

1	ECML-PKDD-15--Taxi-Trajectory-Prediction-I	1
2	Namespace Index	3
2.1	Namespace List	3
3	Hierarchical Index	5
3.1	Class Hierarchy	5
4	Class Index	7
4.1	Class List	7
5	Namespace Documentation	9
5.1	Taxi Namespace Reference	9
5.2	Taxi.Properties Namespace Reference	10
6	Class Documentation	11
6.1	Taxi.Ball Struct Reference	11
6.1.1	Detailed Description	11
6.1.2	Constructor & Destructor Documentation	11
6.1.2.1	Ball(double x, double y, double radius)	11
6.1.3	Property Documentation	12
6.1.3.1	Radius	12
6.1.3.2	X	12
6.1.3.3	Y	12
6.2	Taxi.Cover Class Reference	12
6.2.1	Detailed Description	12
6.2.2	Constructor & Destructor Documentation	12
6.2.2.1	Cover(string csvFilePath)	12
6.2.3	Member Function Documentation	13
6.2.3.1	WhoContains(Polyline polyline)	13
6.2.4	Property Documentation	13
6.2.4.1	Names	13
6.3	Taxi.LearningParameters Class Reference	13

6.3.1	Detailed Description	13
6.3.2	Constructor & Destructor Documentation	14
6.3.2.1	LearningParameters(string learningString)	14
6.3.3	Member Data Documentation	14
6.3.3.1	DispersionExponent	14
6.3.3.2	Keyword	14
6.3.3.3	MaxOccurences	14
6.3.3.4	MinOccurences	14
6.3.3.5	SizeExponent	14
6.4	Taxi.MainForm Class Reference	14
6.4.1	Detailed Description	15
6.4.2	Constructor & Destructor Documentation	15
6.4.2.1	MainForm()	15
6.4.3	Member Function Documentation	15
6.4.3.1	Dispose(bool disposing)	15
6.5	Taxi.Polyline Class Reference	15
6.5.1	Detailed Description	16
6.5.2	Constructor & Destructor Documentation	16
6.5.2.1	Polyline(string line, double partToKeep=1.1f)	16
6.5.3	Member Function Documentation	16
6.5.3.1	Crosses(Ball b)	16
6.5.3.2	FirstElementString(int precision)	16
6.5.3.3	GetDirection()	16
6.5.3.4	LastElementString()	17
6.5.3.5	LastElementString(int precision)	17
6.5.3.6	MaxSpeed()	17
6.5.3.7	Speed()	17
6.5.4	Property Documentation	17
6.5.4.1	Length	17
6.6	Taxi.StreamingCloud Class Reference	17
6.6.1	Detailed Description	18
6.6.2	Member Function Documentation	18
6.6.2.1	Add(WeightedPoint point)	18
6.6.2.2	ToString()	18
6.6.3	Property Documentation	18
6.6.3.1	Barycenter	18
6.6.3.2	Dispersion	18
6.6.3.3	Size	18
6.7	Taxi.StreamingLearning Class Reference	18
6.7.1	Detailed Description	19

6.7.2	Member Function Documentation	19
6.7.2.1	CrossLearning(string filePath, int nFolds, LearningParameters learningParameters)	19
6.7.2.2	CrossValidationScore(string filePath, LearningParameters learningParameters, int nFolds, DistanceFunction distance)	19
6.7.2.3	DistanceFunction(WeightedPoint p1, WeightedPoint p2)	19
6.7.2.4	Train(string filePath, LearningParameters learningParameters)	20
6.7.2.5	TrainPredictAndWrite(string trainFilePath, string testFilePath, string outFilePath, string sampleSubmission, LearningParameters learningParameters)	20
6.8	Taxi.WeightedPoint Class Reference	20
6.8.1	Detailed Description	21
6.8.2	Constructor & Destructor Documentation	21
6.8.2.1	WeightedPoint(double x, double y, double weight)	21
6.8.2.2	WeightedPoint()	21
6.8.2.3	WeightedPoint(WeightedPoint wp)	21
6.8.2.4	WeightedPoint(string line)	21
6.8.3	Member Function Documentation	22
6.8.3.1	Add(WeightedPoint p2)	22
6.8.3.2	Barycenter(ICollection< WeightedPoint > weightedPoints)	22
6.8.3.3	Divide(double lambda)	22
6.8.3.4	Multiply(double lambda)	22
6.8.3.5	ToString()	23
6.8.4	Property Documentation	23
6.8.4.1	Weight	23
6.8.4.2	X	23
6.8.4.3	Y	23
	Index	25

Chapter 1

ECML-PKDD-15--Taxi-Trajectory-Prediction-I

This is the program I used when taking part in the ECML-PKDD-15–Taxi-Trajectory-Prediction-I.

The underlying model is very simple and the program was designed so that the learning (on-line) and prediction steps are very fast. A more detailed explanation can be found in the .pdf

I achieved 17th / 381.

Possible improvements include Bag-Of-Words, decision trees approach...

Chapter 2

Namespace Index

2.1 Namespace List

Here is a list of all documented namespaces with brief descriptions:

Taxi	9
Taxi.Properties	10

Chapter 3

Hierarchical Index

3.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Taxi.Ball	11
Taxi.Cover	12
Form	
Taxi.MainForm	14
Taxi.LearningParameters	13
Taxi.Polyline	15
Taxi.StreamingCloud	17
Taxi.StreamingLearning	18
Taxi.WeightedPoint	20

Chapter 4

Class Index

4.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Taxi.Ball	Representation of a ball : (x, y) represents the center, r the radius.	11
Taxi.Cover	Represents a cover of the map. Basically, balls centers and radiuses. Balls are named after there center and radiuses.	12
Taxi.LearningParameters	A simple class to store learning parameters	13
Taxi.MainForm	Main form of the application.	14
Taxi.Polyline	Represents a polyline : $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ Various features are available.	15
Taxi.StreamingCloud	Represents a cloud of points. Note that the points themselves are not stored.	17
Taxi.StreamingLearning	Learning methods.	18
Taxi.WeightedPoint	Points with a mass : (x, y, m)	20

Chapter 5

Namespace Documentation

5.1 Taxi Namespace Reference

Namespaces

- namespace [Properties](#)

Classes

- struct [Ball](#)
Representation of a ball : (x, y) represents the center, r the radius.
- class [Cover](#)
Represents a cover of the map. Basically, balls centers and radiuses. Balls are named after there center and radiuses.
- class **DataCleaner**
A static class that cleans the data.
- class **Distances**
Implementation of metrics over (weighted) points.
- class **Extension**
Simple dictionary extension for Dictionary string [StreamingCloud](#).
- class **FeatureInteractions**
Implements various interactions to extract relevant features.
- class **FeatureWriter**
Turns the data into a "libsvm" representation.
- class [LearningParameters](#)
A simple class to store learning parameters
- class [MainForm](#)
Main form of the application.
- class [Polyline](#)
Represents a polyline : $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ Various features are available.
- class **Program**
- class [StreamingCloud](#)
Represents a cloud of points. Note that the points themselves are not stored.
- class [StreamingLearning](#)
Learning methods.
- class [WeightedPoint](#)
Points with a mass : (x, y, m) .

5.2 Taxi.Properties Namespace Reference

Classes

- class **Resources**

Une classe de ressource fortement typée destinée, entre autres, à la consultation des chaînes localisées.

- class **Settings**

Chapter 6

Class Documentation

6.1 Taxi.Ball Struct Reference

Representation of a ball : (x, y) represents the center, r the radius.

Public Member Functions

- **Ball** (double x , double y , double $radius$)
Constructs a ball, centered in (x, y) with radius r .

Properties

- double **X** [get]
 x accessor
- double **Y** [get]
 y accessor
- double **Radius** [get]
 r accessor

6.1.1 Detailed Description

Representation of a ball : (x, y) represents the center, r the radius.

6.1.2 Constructor & Destructor Documentation

6.1.2.1 Taxi.Ball.Ball (double x , double y , double $radius$) [inline]

Constructs a ball, centered in (x, y) with radius r .

Parameters

x	x
y	y
$radius$	r

6.1.3 Property Documentation

6.1.3.1 double Taxi.Ball.Radius [get]

r accessor

6.1.3.2 double Taxi.Ball.X [get]

x accessor

6.1.3.3 double Taxi.Ball.Y [get]

y accessor

The documentation for this struct was generated from the following file:

- Taxi/Geometry/Ball.cs

6.2 Taxi.Cover Class Reference

Represents a cover of the map. Basically, balls centers and radiuses. Balls are named after there center and radiuses.

Public Member Functions

- [Cover](#) (string csvFilePath)
Builds a cover from a .csv file.
- List< string > [WhoContains](#) ([Polyline](#) polyline)
Returns the names of the balls containing the specified polyline

Properties

- List< string > [Names](#) [get]
Accessor to the names of the balls in the cover.

6.2.1 Detailed Description

Represents a cover of the map. Basically, balls centers and radiuses. Balls are named after there center and radiuses.

6.2.2 Constructor & Destructor Documentation

6.2.2.1 Taxi.Cover.Cover (string csvFilePath) [inline]

Builds a cover from a .csv file.

Parameters

<i>csvFilePath</i>	.csv file containing the cover. Must contain a header.
--------------------	--

6.2.3 Member Function Documentation

6.2.3.1 List<string> Taxi.Cover.WhoContains (Polyline polyline) [inline]

Returns the names of the balls containing the specified polyline

Parameters

<i>polyline</i>	The polyline to study
-----------------	-----------------------

Returns

The names of the balls crossing the polyline

6.2.4 Property Documentation

6.2.4.1 List<string> Taxi.Cover.Names [get]

Accessor to the names of the balls in the cover.

The documentation for this class was generated from the following file:

- Taxi/Geometry/Cover.cs

6.3 Taxi.LearningParameters Class Reference

A simple class to store learning parameters

Public Member Functions

- [LearningParameters](#) (string learningString)
Constructs the object from its string representation.

Public Attributes

- int [MinOccurences](#)
The smallest number of occurences of a feature to use it.
- int [MaxOccurences](#)
The largest number of occurences of a feature to use it.
- double [DispersionExponent](#)
The exponent weighting the dispersion of the cloud.
- double [SizeExponent](#)
The exponent weighting the size of the cloud.
- string [Keyword](#)
The method used to generate combination of features.

6.3.1 Detailed Description

A simple class to store learning parameters

6.3.2 Constructor & Destructor Documentation

6.3.2.1 Taxi.LearningParameters.LearningParameters (string *learningString*) [inline]

Constructs the object from its string representation.

Parameters

<i>learningString</i>	"Keyword_MinOccurences_MaxOccurences_SizeExponent_DispersionExponent"
-----------------------	---

6.3.3 Member Data Documentation

6.3.3.1 double Taxi.LearningParameters.DispersionExponent

The exponent weighting the dispersion of the cloud.

6.3.3.2 string Taxi.LearningParameters.Keyword

The method used to generate combination of features.

6.3.3.3 int Taxi.LearningParameters.MaxOccurences

The largest number of occurences of a feature to use it.

6.3.3.4 int Taxi.LearningParameters.MinOccurences

The smallest number of occurences of a feature to use it.

6.3.3.5 double Taxi.LearningParameters.SizeExponent

The exponent weighting the size of the cloud.

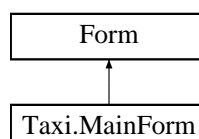
The documentation for this class was generated from the following file:

- Taxi/Learning/PredictionParameters.cs

6.4 Taxi.MainForm Class Reference

Main form of the application.

Inheritance diagram for Taxi.MainForm:



Public Member Functions

- [MainForm \(\)](#)

Initialization of the main form of the application.

Protected Member Functions

- override void [Dispose](#) (bool disposing)
Clean up any resources being used.

6.4.1 Detailed Description

Main form of the application.

6.4.2 Constructor & Destructor Documentation

6.4.2.1 Taxi.MainForm.MainForm () [inline]

Initialization of the main form of the application.

6.4.3 Member Function Documentation

6.4.3.1 override void Taxi.MainForm.Dispose (bool disposing) [inline],[protected]

Clean up any resources being used.

Parameters

<i>disposing</i>	true if managed resources should be disposed; otherwise, false.
------------------	---

The documentation for this class was generated from the following files:

- Taxi/MainForm.cs
- Taxi/MainForm.Designer.cs

6.5 Taxi.Polyline Class Reference

Represents a polyline : $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ Various features are available.

Public Member Functions

- [Polyline](#) (string line, double partToKeep=1.1f)
Builds a polyline from a string. Optional : keep a subpart of the complete polyline.
- string [GetDirection](#) ()
Feature extraction. Direction of the polyline : N,S,E,W.
- double [MaxSpeed](#) ()
Feature extraction.
- bool [Crosses](#) (Ball b)
Feature extraction. True if the polyline goes through the ball.
- string [LastElementString](#) ()
Feature extraction. Returns (as a string) the last point of a polyline.
- string [LastElementString](#) (int precision)
Feature extraction. Returns (as a string) the last point of a polyline.
- string [FirstElementString](#) (int precision)
Feature extraction. Returns (as a string) the first point of a polyline.
- double [Speed](#) ()
Feature extraction. Gets the average speed.

Properties

- int `Length` `[get]`
Length of the polyline.

6.5.1 Detailed Description

Represents a polyline : $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ Various features are available.

6.5.2 Constructor & Destructor Documentation

6.5.2.1 `Taxi.Polyline.Polyline (string line, double partToKeep = 1.1f) [inline]`

Builds a polyline from a string. Optional : keep a subpart of the complete polyline.

Parameters

<i>line</i>	The input string, under the format <code>[[x_1,y_1],...,[x_n,y_n]]</code> .
<i>partToKeep</i>	The percentage of the first part of the string to keep.

6.5.3 Member Function Documentation

6.5.3.1 `bool Taxi.Polyline.Crosses (Ball b) [inline]`

Feature extraction. True if the polyline goes through the ball.

Parameters

<i>b</i>	The ball.
----------	-----------

Returns

True if the polyline crosses the ball.

6.5.3.2 `string Taxi.Polyline.FirstElementString (int precision) [inline]`

Feature extraction. Returns (as a string) the first point of a polyline.

Returns

The (rounded) first point of a polyline

6.5.3.3 `string Taxi.Polyline.GetDirection () [inline]`

Feature extraction. Direction of the polyline : N,S,E,W.

Returns

The direction, as a string, of the polyline.

6.5.3.4 string Taxi.Polyline.LastElementString () [inline]

Feature extraction. Returns (as a string) the last point of a polyline.

Returns

The last point of a polyline

6.5.3.5 string Taxi.Polyline.LastElementString (int *precision*) [inline]

Feature extraction. Returns (as a string) the last point of a polyline.

Returns

The (rounded) last point of a polyline.

6.5.3.6 double Taxi.Polyline.MaxSpeed () [inline]

Feature extraction.

Returns

The maximum speed over the polyline.

6.5.3.7 double Taxi.Polyline.Speed () [inline]

Feature extraction. Gets the average speed.

Returns

The average speed of the trajectory.

6.5.4 Property Documentation

6.5.4.1 int Taxi.Polyline.Length [get]

Length of the polyline.

The documentation for this class was generated from the following file:

- Taxi/Geometry/Polyline.cs

6.6 Taxi.StreamingCloud Class Reference

Represents a cloud of points. Note that the points themselves are not stored.

Public Member Functions

- void [Add](#) ([WeightedPoint](#) point)
Adds a point to a streaming cloud : the mean and variance of the clouds are updated.
- override string [ToString](#) ()
String representation of a cloud : barycenter, dispersion and the number of points.

Properties

- [WeightedPoint Barycenter](#) [get]
Streaming estimate of the barycenter of the cloud.
- [double Dispersion](#) [get]
Streaming estimate of the variance of the cloud.
- [int Size](#) [get]
Size of the cloud : /f\$/f\$.

6.6.1 Detailed Description

Represents a cloud of points. Note that the points themselves are not stored.

6.6.2 Member Function Documentation

6.6.2.1 void Taxi.StreamingCloud.Add ([WeightedPoint point](#)) [inline]

Adds a point to a streaming cloud : the mean and variance of the clouds are updated.

Parameters

<i>point</i>	The point to add to the cloud.
--------------	--------------------------------

6.6.2.2 override string Taxi.StreamingCloud.ToString () [inline]

String representation of a cloud : barycenter, dispersion and the number of points.

Returns

"barycenter_dispersion_nbPoints"

6.6.3 Property Documentation

6.6.3.1 [WeightedPoint Taxi.StreamingCloud.Barycenter](#) [get]

Streaming estimate of the barycenter of the cloud.

6.6.3.2 [double Taxi.StreamingCloud.Dispersion](#) [get]

Streaming estimate of the variance of the cloud.

6.6.3.3 [int Taxi.StreamingCloud.Size](#) [get]

Size of the cloud : /f\$/f\$.

The documentation for this class was generated from the following file:

- Taxi/Geometry/StreamingCloud.cs

6.7 Taxi.StreamingLearning Class Reference

Learning methods.

Public Member Functions

- delegate double [DistanceFunction](#) ([WeightedPoint](#) p1, [WeightedPoint](#) p2)
The distance function.
- void [CrossLearning](#) (string filePath, int nFolds, [LearningParameters](#) learningParameters)
Learns over each fold.
- double[] [CrossValidationScore](#) (string filePath, [LearningParameters](#) learningParameters, int nFolds, [DistanceFunction](#) distance)
Evaluates the cross validation error for each fold.
- void [Train](#) (string filePath, [LearningParameters](#) learningParameters)
Trains the model.
- void [TrainPredictAndWrite](#) (string trainFilePath, string testFilePath, string outFilePath, string sample↔Submission, [LearningParameters](#) learningParameters)
Trains the model and generate the predictions.

6.7.1 Detailed Description

Learning methods.

6.7.2 Member Function Documentation

6.7.2.1 void Taxi.StreamingLearning.CrossLearning (string filePath, int nFolds, [LearningParameters](#) learningParameters)
[inline]

Learns over each fold.

Parameters

<i>filePath</i>	The training file path (after feature extraction).
<i>nFolds</i>	The number of folds.
<i>learning↔Parameters</i>	The learning parameters.

6.7.2.2 double [] Taxi.StreamingLearning.CrossValidationScore (string filePath, [LearningParameters](#) learningParameters, int nFolds, [DistanceFunction](#) distance) [inline]

Evaluates the cross validation error for each fold.

Parameters

<i>filePath</i>	The training file path (after feature extraction).
<i>learning↔Parameters</i>	The learning parameters.
<i>nFolds</i>	The number of folds.
<i>distance</i>	The distance function to evaluate the scores.

Returns

An array, each element containing the error over the fold.

6.7.2.3 delegate double Taxi.StreamingLearning.DistanceFunction ([WeightedPoint](#) p1, [WeightedPoint](#) p2)

The distance function.

Parameters

<i>p1</i>	First point.
<i>p2</i>	Second point.

Returns

The distance between p1 and p2

6.7.2.4 void Taxi.StreamingLearning.Train (string *filePath*, LearningParameters *learningParameters*) [inline]

Trains the model.

Parameters

<i>filePath</i>	The training file path (after feature extraction).
<i>learningParameters</i>	The learning parameters.

6.7.2.5 void Taxi.StreamingLearning.TrainPredictAndWrite (string *trainFilePath*, string *testFilePath*, string *outFilePath*, string *sampleSubmission*, LearningParameters *learningParameters*) [inline]

Trains the model and generate the predictions.

Parameters

<i>trainFilePath</i>	The training file path (after feature extraction).
<i>testFilePath</i>	The testing file path (after feature extraction).
<i>outFilePath</i>	The predicted values file path.
<i>sampleSubmission</i>	The sample submission file path (as provided by Kaggle).
<i>learningParameters</i>	The learning parameters.

The documentation for this class was generated from the following file:

- Taxi/Learning/StreamingLearning.cs

6.8 Taxi.WeightedPoint Class Reference

Points with a mass : (x, y, m) .

Public Member Functions

- [WeightedPoint](#) (double x, double y, double weight)
Constructs a point with a mass (x, y, m) .
- [WeightedPoint](#) ()
Creates a null weighted point.
- [WeightedPoint](#) ([WeightedPoint](#) wp)
(Deep) copy constructor of a weighted point.
- [WeightedPoint](#) (string line)
From string constructor of a weighted point.
- [WeightedPoint Add](#) ([WeightedPoint](#) p2)
Performs $(x_1 + x_2, y_1 + y_2, m_1 + m_2)$.

- [WeightedPoint Divide](#) (double lambda)
Performs $(x_1/\lambda, y_1/\lambda, m_1/\lambda)$
- [WeightedPoint Multiply](#) (double lambda)
Performs $(\lambda x_1, \lambda y_1, \lambda m_1)$.
- new string [ToString](#) ()
Represents the position of the weighted point as a string.

Static Public Member Functions

- static [WeightedPoint Barycenter](#) (IList< [WeightedPoint](#) > weightedPoints)
Given a list of WeightedPoints returns.

Properties

- double [X](#) [get]
Returns x
- double [Y](#) [get]
Returns y
- double [Weight](#) [get, set]
Returns m

6.8.1 Detailed Description

Points with a mass : (x, y, m) .

6.8.2 Constructor & Destructor Documentation

6.8.2.1 Taxi.WeightedPoint.WeightedPoint (double x , double y , double $weight$) [inline]

Constructs a point with a mass (x, y, m) .

Parameters

x	
y	
$weight$	

6.8.2.2 Taxi.WeightedPoint.WeightedPoint () [inline]

Creates a null weighted point.

6.8.2.3 Taxi.WeightedPoint.WeightedPoint ([WeightedPoint](#) wp) [inline]

(Deep) copy constructor of a weighted point.

Parameters

wp	Weighted point to copy.
------	-------------------------

6.8.2.4 Taxi.WeightedPoint.WeightedPoint (string $line$) [inline]

From string constructor of a weighted point.

Parameters

<i>line</i>	String "x_y"
-------------	--------------

6.8.3 Member Function Documentation

6.8.3.1 **WeightedPoint** Taxi.**WeightedPoint**.Add (**WeightedPoint** *p2*) [inline]

Performs $(x_1 + x_2, y_1 + y_2, m_1 + m_2)$.

Parameters

<i>p2</i>	The point to add.
-----------	-------------------

Returns

$(x_1+x_2, y_1+y_2, m_1+m_2)$.

6.8.3.2 **static WeightedPoint** Taxi.**WeightedPoint**.Barycenter (**IList**< **WeightedPoint** > *weightedPoints*) [inline],
[static]

Given a list of WeightedPoints returns.

Parameters

<i>weightedPoints</i>	Weighted points to average.
-----------------------	-----------------------------

Returns

The barycenter of the input points.

6.8.3.3 **WeightedPoint** Taxi.**WeightedPoint**.Divide (**double** *lambda*) [inline]

Performs $(x_1/\lambda, y_1/\lambda, m_1/\lambda)$

Parameters

<i>lambda</i>	The division coefficient.
---------------	---------------------------

Returns

The scaled weighted point.

6.8.3.4 **WeightedPoint** Taxi.**WeightedPoint**.Multiply (**double** *lambda*) [inline]

Performs $(\lambda x_1, \lambda y_1, \lambda m_1)$.

Parameters

<i>lambda</i>	The multiplication coefficient.
---------------	---------------------------------

Returns

The scaled weighted point.

6.8.3.5 `new string Taxi.WeightedPoint.ToString () [inline]`

Represents the position of the weighted point as a string.

Returns

"x_y"

6.8.4 Property Documentation

6.8.4.1 `double Taxi.WeightedPoint.Weight [get], [set]`

Returns *m*

6.8.4.2 `double Taxi.WeightedPoint.X [get]`

Returns *x*

6.8.4.3 `double Taxi.WeightedPoint.Y [get]`

Returns *y*

The documentation for this class was generated from the following file:

- Taxi/Geometry/WeightedPoint.cs

Index

Add

Taxi::StreamingCloud, [18](#)
Taxi::WeightedPoint, [22](#)

Ball

Taxi::Ball, [11](#)

Barycenter

Taxi::StreamingCloud, [18](#)
Taxi::WeightedPoint, [22](#)

Cover

Taxi::Cover, [12](#)

CrossLearning

Taxi::StreamingLearning, [19](#)

CrossValidationScore

Taxi::StreamingLearning, [19](#)

Crosses

Taxi::Polyline, [16](#)

Dispersion

Taxi::StreamingCloud, [18](#)

DispersionExponent

Taxi::LearningParameters, [14](#)

Dispose

Taxi::MainForm, [15](#)

DistanceFunction

Taxi::StreamingLearning, [19](#)

Divide

Taxi::WeightedPoint, [22](#)

FirstElementString

Taxi::Polyline, [16](#)

GetDirection

Taxi::Polyline, [16](#)

Keyword

Taxi::LearningParameters, [14](#)

LastElementString

Taxi::Polyline, [16](#), [17](#)

LearningParameters

Taxi::LearningParameters, [14](#)

Length

Taxi::Polyline, [17](#)

MainForm

Taxi::MainForm, [15](#)

MaxOccurrences

Taxi::LearningParameters, [14](#)

MaxSpeed

Taxi::Polyline, [17](#)

MinOccurrences

Taxi::LearningParameters, [14](#)

Multiply

Taxi::WeightedPoint, [22](#)

Names

Taxi::Cover, [13](#)

Polyline

Taxi::Polyline, [16](#)

Radius

Taxi::Ball, [12](#)

Size

Taxi::StreamingCloud, [18](#)

SizeExponent

Taxi::LearningParameters, [14](#)

Speed

Taxi::Polyline, [17](#)

Taxi, [9](#)

Taxi.Ball, [11](#)

Taxi.Cover, [12](#)

Taxi.LearningParameters, [13](#)

Taxi.MainForm, [14](#)

Taxi.Polyline, [15](#)

Taxi.Properties, [10](#)

Taxi.StreamingCloud, [17](#)

Taxi.StreamingLearning, [18](#)

Taxi.WeightPoint, [20](#)

Taxi::Ball

Ball, [11](#)

Radius, [12](#)

X, [12](#)

Y, [12](#)

Taxi::Cover

Cover, [12](#)

Names, [13](#)

WhoContains, [13](#)

Taxi::LearningParameters

DispersionExponent, [14](#)

Keyword, [14](#)

LearningParameters, [14](#)

MaxOccurrences, [14](#)

MinOccurrences, [14](#)

SizeExponent, [14](#)

Taxi::MainForm

Dispose, [15](#)

MainForm, [15](#)

- Taxi::Polyline
 - Crosses, [16](#)
 - FirstElementString, [16](#)
 - GetDirection, [16](#)
 - LastElementString, [16](#), [17](#)
 - Length, [17](#)
 - MaxSpeed, [17](#)
 - Polyline, [16](#)
 - Speed, [17](#)
- Taxi::StreamingCloud
 - Add, [18](#)
 - Barycenter, [18](#)
 - Dispersion, [18](#)
 - Size, [18](#)
 - ToString, [18](#)
- Taxi::StreamingLearning
 - CrossLearning, [19](#)
 - CrossValidationScore, [19](#)
 - DistanceFunction, [19](#)
 - Train, [20](#)
 - TrainPredictAndWrite, [20](#)
- Taxi::WeightedPoint
 - Add, [22](#)
 - Barycenter, [22](#)
 - Divide, [22](#)
 - Multiply, [22](#)
 - ToString, [22](#)
 - Weight, [23](#)
 - WeightedPoint, [21](#)
 - X, [23](#)
 - Y, [23](#)
- ToString
 - Taxi::StreamingCloud, [18](#)
 - Taxi::WeightedPoint, [22](#)
- Train
 - Taxi::StreamingLearning, [20](#)
- TrainPredictAndWrite
 - Taxi::StreamingLearning, [20](#)
- Weight
 - Taxi::WeightedPoint, [23](#)
- WeightedPoint
 - Taxi::WeightedPoint, [21](#)
- WhoContains
 - Taxi::Cover, [13](#)
- X
 - Taxi::Ball, [12](#)
 - Taxi::WeightedPoint, [23](#)
- Y
 - Taxi::Ball, [12](#)
 - Taxi::WeightedPoint, [23](#)