# WSO2 Machine Learner 1.1.0 - Alpha Release

The alpha release of WSO2 Machine Learner (ML) 1.1.0 comes with the recommendation algorithm support. Recommendation solutions are very useful for retail websites to predict a rating or a preference, a user would give to an item.

WSO2 ML 1.1.0 release consists of following features;

- Deep learning algorithm support
- Anomaly detection algorithm support
- Recommendation algorithm support
- PMML support

For general information on WSO2 Machine Learner 1.1.0-alpha release, please visit our documentation <u>https://docs.wso2.com/display/ML110/WSO2+Machine+Learner+Documentation</u>

Collaborative Filtering Algorithm Steps for Building a Collaborative Filtering Model with explicit data using WSO2 ML Step 1 - Create an Analysis Step 2 - Algorithm Selection Step 3 - Hyper Parameters Step 4 - Model Building Steps for Building a Collaborative Filtering Model with implicit feedback data using WSO2 ML Step 1 - Create an Analysis Step 2 - Algorithm Selection Step 3 - Hyper Parameters Step 4 - Model Building Collaborative Filtering for WSO2 ML - Samples Generating a Model Using the Collaborative Filtering for explicit data Algorithm Introduction Prerequisites Executing the sample Output of the sample Viewing the model Stacked Autoencoders Algorithm Steps for Building a Deep Learning Model using WSO2 ML Step 1 - Create an Analysis Step 2 - Algorithm Selection Step 3 - Hyper Parameters Step 4 - Model Building

Step 6 - Prediction Deep Learning for WSO2 ML - Samples Generating a Model Using the Stacked Autoencoders Algorithm Introduction Prerequisites Executing the sample Output of the sample Viewing the model Anomaly Detection Algorithm Steps for Building an Anomaly Detection Model using WSO2 ML Step 1 - Create an Analysis Step 2 - Algorithm Selection Step 3 - Hyper Parameters Step 4 - Model Building Step 5 - Model Summary Step 6 - Prediction Anomaly Detection for WSO2 ML - Samples Generating a Model Using the K Means Anomaly Detection Algorithm with unlabeled data Introduction Prerequisites Executing the sample Output of the sample Viewing the model Generating a Model Using the K Means Anomaly Detection Algorithm with labeled data Introduction Prerequisites Executing the sample Output of the sample Viewing the model Viewing the model prediction Generating a Tuned Model Using the K Means Anomaly Detection Algorithm with labeled data Introduction Prerequisites Executing the sample Output of the sample Viewing the model summary Viewing the model prediction PMML Support PMML PMML Support in WSO2 ML Usage

Downloading a model in PMML Format <u>Method 1</u> <u>Method 2</u> <u>Publishing a model to Registry in PMML Format</u> <u>Method 1</u> <u>Method 2</u>

#### <u>APIs</u>

Finding whether an algorithm supports an export type Overview Parameter description Sample cURL command Example Sample output **REST API response** Export a model <u>Overview</u> Parameter description Sample cURL command Example Sample output **REST API response** Publish a model <u>Overview</u> Parameter description Sample cURL command Example Sample output **REST API response** Samples

# **Collaborative Filtering Algorithm**

Collaborative filtering is commonly used for recommender systems. These techniques aim to fill in the missing entries of a user-item association matrix. MLlib currently supports model-based collaborative filtering, in which users and products are described by a small set of latent factors that can be used to predict missing entries. MLlib uses the alternating least squares (ALS) algorithm to learn these latent factors.

There are two scenarios to consider in collaborative filtering.

- 1. Explicit entries in the user-item matrix as *explicit* preferences(ratings) given by the user to the item
- 2. Implicit feedback Preferences on products are implicit feedbacks such as views, clicks, purchases, likes, shares etc.

WSO2 Machine Learner supports both of these two scenarios.

# Steps for Building a Collaborative Filtering Model with explicit data using WSO2 ML

#### Step 1 - Create an Analysis

Upload the dataset which has explicit data. A dataset should contain a user ID column, a product ID column and a ratings column. After creating the dataset, create a project from that dataset.

Then start a new analysis to build a collaborative filtering model.

⊙ Collaborative-Filtering-Project	[ created: 2015-11-30 12:59:48.78 ] 🛛 🥖	No analyses available	DELETE PROJECT
Collaborative filtering with ratings dataset	Analysis name <b>O</b> aborative-filtering-analysis		

#### Step 2 - Algorithm Selection

In the Algorithm selection process there is a new category called Recommendation. Select Collaborative Filtering (Explicit Data) under that category.

Step 1 Preprocess	<sup>Step 2</sup> Explore	<sub>Step 3</sub> Algorithms	<sub>Step</sub> 4 Parameters	<sub>Step</sub> 5 Model
Algo	rithm			
Algorithi	m name *			
COLLA	ABORATIVE FILTERING (E	xplicit Data)		•
Train da	ta fraction *			
0.7				
User var	iable *			
USER	_ID			•
Product	variable *			
PROD	UCT_ID			•
Rating v	ariable *			
RATIN	IG			•

At the algorithm selection, you will have to specify which columns represent user variable, product variable and the rating variable. Select those entries from the column names of the dataset.

#### Step 3 - Hyper Parameters

In the parameter selection step you have to input necessary hyper parameters for the model.

- Rank Number of latent factors in the model.
- Iterations Number of iterations in the Alternative Least Squares computation.
- Lambda Regularization parameter in Alternative Least Squares computation.
- Blocks Level of parallelism to split the computation into.

<sub>Step 1</sub> Preprocess	Step 2 Explo	step 3 Algo	rithms	<sub>Step</sub> 4 Parameters	<sub>Step 5</sub> Model
Pa	arameters				
Set	t Hyper-Parameters f	or Recommendation \ C	OLLABORATIVE FILT	TERING	
8					
Iter	rations 😡				
20	0				
Lan	mbda 😧				
0.	.01				
Blo	ocks 😮				
-1	1				

# Step 4 - Model Building

Then after selecting the dataset version you can build the model.



Step 5 - Model Summary

After successfully building the model you can view the model summary. Model summary will provide you the mean squared error of the build model.

# Model Summary [MSE: 6.03e-1]

# Steps for Building a Collaborative Filtering Model with implicit feedback data using WSO2 ML

## Step 1 - Create an Analysis

Upload the dataset which has explicit data. A dataset should contain a user ID column, a product ID column, any other observations such as page views, purchases, time spent in page, recurrent visits etc. After creating the dataset, create a project from that dataset. Then start a new analysis to build a collaborative filtering model.

🛇 collaborative-filtering-implicit-feedback-project 🛛 [created: 2015-11-30 14:07:04.786] 🛕 No analyses available		COMPARE MODELS
implicit-feedback	Analysis name  kollaborative-filtering-impl CREATE ANALYSIS	

## Step 2 - Algorithm Selection

In the Algorithm selection process there is a new category called Recommendation. Select Collaborative Filtering (Implicit Feedback Data) under that category.

<sub>Step 1</sub> Preprocess	<sub>Step 2</sub> Explore	<sub>Step 3</sub> Algorithms	<sub>Step</sub> 4 Parameters	<sub>Step 5</sub> Model	
Algo	rithm				
Algorith	m name *				
COLD	ABORATIVE FILTERING (Impl	icit Feedback Data)		•	
Train da	ta fraction *				
0.7					
User var	iable *				
USER	_ID			<u> </u>	
Product	variable *				
PROD	UCT_ID			•	
Observa	ition list				
2,3					

At the algorithm selection, you will have to specify which columns represent user variable, product variable and the the column number of observations. Select those entries from the column names of the dataset.

#### Step 3 - Hyper Parameters

In the parameter selection step you have to input necessary hyper parameters for the model.

- Rank Number of latent factors in the model.
- Iterations Number of iterations in the Alternative Least Squares computation.
- Lambda Regularization parameter in Alternative Least Squares computation.
- Blocks Level of parallelism to split the computation into.
- Alpha Confidence parameter.
- Weights Comma separated weights given to observation fields.

Step 1 Preprocess	<sub>Step 2</sub> Explore	<sub>Step 3</sub> Algorithms	<sub>Step</sub> 4 Parameters	Step 5 Model	
Para	meters				
Set Hype	r-Parameters for Recom	mendation\ COLLABORATIV	FILTERING IMPLICIT		

Rank 😧	
8	
Iterations 😧	
20	
Lambda 😧	
0.01	
Blocks 🚱	
-1	
Alpha 🔞	
40	
Weights 9	
30,70	

•

# Step 4 - Model Building

Then after selecting the dataset version you can build the model.

# Model

Dataset version

Recommendation-implicit-feedback-1.0.0

#### Step 5 - Model Summary

After successfully building the model you can view the model summary. Model summary will provide you the mean squared error of the build model.

# Model Summary [MSE: 3.14e+1]

# Collaborative Filtering for WSO2 ML - Samples

Generating a Model Using the Collaborative Filtering for explicit data Algorithm

- Introduction
- Prerequisites
- Executing the sample
- Output of the sample

#### Introduction

This sample demonstrates how a model is generated out of a data set using the collaborative filtering for explicit data algorithm.

#### Prerequisites

Follow the steps below to set up the prerequisites before you start.

- 1. Download WSO2 Machine Learner, and start the server. For information on setting up and running WSO2 ML, see <u>Getting Started</u>.
- 2. Download and install jq (CLI JSON processor). For instructions, see jq Documentation.
- 3. If you are using Mac OS X, download and install GNU stream editor (sed). For instructions, see <u>GNU sed Documentation</u>.

#### **Executing the sample**

Follow the steps below to execute the sample.

- 1. Navigate to <ML\_HOME>/samples/default/collaborative-filtering/ directory using the CLI.
- 2. Execute the following command to execute the sample: ./model-generation.sh

#### Output of the sample

Once the sample is successfully executed, you can view the prediction of the model. By default , the sample generates the model in the <ML\_HOME>/models/ directory of your machine. For an example, the generated file is in the following format denoting the date and time when it was generated:

wso2-ml-collaborative-filtering-explicit-sample-analysis.Model.2015-11-26\_12-00-46

#### Viewing the model

You can view the summary of the built model using the ML UI as follows.

- Log in to the ML UI from your Web browser using admin/admin credentials and the following URL: https://<ML\_HOST>:<ML\_PORT>/ml
- 2. Click the Projects button as shown below.



3. Click MODELS button of the new analysis which you created by executing the sample as shown below.

Swso2-ml-collaborative-filtering-ex		
This project tests ml workflow for collaborative filtering explicit model	Analysis name 🛛	
	e.g. myanalysis   CREATE ANALYSIS	
	so2-ml-collaborative-filtering-explicit-sample-analysis	(b) VIEW (c) MODELS (c) DELETE

4. You view the built new model as shown below.

ura2 mi callabarativa filtaring avalicit cample analysic Madel 2015 11 20 17 20 25	[Crasted: 2015, 11, 20, 17:28:25, 008, 1	D VIEW PREDICT		
[Dataset version: undefined] [Status: Complete w]	[ ereated2013-11-30 17-30-330 ]		(iii) (	DELETE MODEL

#### Viewing the recommendations

The sample generates recommendations from the model for the given set of users and the products. Product recommendations are [125,123,124] and user recommendations are [1,2]

# Stacked Autoencoders Algorithm

This is a multi-layer feed-forward artificial neural network that is trained with stochastic gradient descent using back-propagation. The network can contain a large number of hidden layers consisting of neurons with tanh, rectifier and maxout activation functions. Subsequent layers learns from activation from previous layers. Each compute node trains a copy of the global model parameters on its local data with multi-threading (asynchronously), and contributes periodically to the global model via model averaging across the network.



This is used as a classifier in ML.

The above diagram shows a deep network with 4 inputs (4 features), 2 hidden layers and 2 outputs (2 classes to be predicted).

For more information on the implementation, please refer to the documentation: <u>Implemented</u> <u>H2O Deeplearning and visualization for WSO2-ML</u>.

# Steps for Building a Deep Learning Model using WSO2 ML

### Step 1 - Create an Analysis

Upload the dataset and create a new project.

As for the every model first you have to upload a dataset and create a new project. Then start a new analysis to build a deep learning model.

	(+) CREATE PROJECT	
O Diabetes_Project	[created: 2015-11-26 08:57:08:51 ] 🛕 No analyses available	COMPARE MODELS () DELETE PROJECT
Diabetes project	Analysis name  wso2-ml-stacked-autoence CREATE ANALYSIS	

### Step 2 - Algorithm Selection

In the Algorithm selection process there is a new category called Deep Learning. Select Stacked Autoencoders under that category.

<sup>Step 1</sup> Preprocess	Step 2 Explore	Step 3 Algorithms	<sub>Step</sub> 4 Parameters	<sub>Step 5</sub> Model
A	gorithm			
Alg	orithm name *			
5	STACKED AUTOENCODERS			•
Res	sponse variable *			
(	Class			-
Tra	in data fraction *			
0.	.7			

### Step 3 - Hyper Parameters

In the parameter selection step you have to input necessary hyper parameters for the model.

- Batch Size Number of training cases considered in each epoch
- Layer Sizes Number of neurons in each layer. First layers is closest to the inputs and the last layer is furthest.
- Activation Type The activation function (non-linearity) to be used for the neurons in the hidden layers. It can be one of the following:
  - Rectifier
  - RectifierWithDropout
  - Tanh
  - TanhWithDropout
  - Maxout
  - MaxoutWithDropout
- Epochs Number of iterations the network is trained.

In addition to choosing hyperparameters, the deep network visualization will be show with the current configuration of parameters. Click "update visualization" if you want to change it after changing parameters.

#### Parameters

Set Hyper-Parameters for Deeplearning\ STACKED AUTOENCODERS

Batch Size 🗿	
100	
Layer Sizes 📀	
500,500,500	
Activation Type	
RectifierWithDropout	
Epochs 📀	
10	

# Deep Network Visualization



## Step 4 - Model Building

Then after selecting the dataset version you can build the model.

Step 1 Preprocess	Step 2 Explore	<sub>Step 3</sub> Algorithms	<sub>Step</sub> 4 Parameters	Step 5 Model
Mode	21			
Dataset v	ersion			
diabete	es_dataset-1.0.0			•

#### Step 5 - Model Summary

After successfully building the model you can view the model summary. In the summary you can get an overall idea about the performance of the build model. Model summary will provide measurements such as accuracy, confusion matrix and predicted vs. actual graph. You will be able to evaluate your model based on these metrics and may choose different parameters for a better model.



Model Summary [Accuracy: 61.16%]

Moreover, you will be able to see the visualization of the deep network in the model summary.



# Step 6 - Prediction

This is where you can predict new data using built model. As the input, you have to give feature values of new data point or you can give new data as a batch using csv or tsv file. So after providing inputs to those values you will get the predictions for new data.

## Predict

Feature values	
NumPregnancies *	
2	
PG2 *	
3	
DBP *	
4	
TSFT *	
5	
SI2 *	
1	
BMI *	
4	
DPF *	
6	
Age *	
7	

# Deep Learning for WSO2 ML - Samples

Generating a Model Using the Stacked Autoencoders Algorithm

- Introduction
- Prerequisites
- Executing the sample
- Output of the sample

### Introduction

This sample demonstrates how a model is generated out of a data set using the stacked autoencoders deep learning algorithm.

#### Prerequisites

Follow the steps below to set up the prerequisites before you start.

- 1. Download WSO2 Machine Learner, and start the server. For information on setting up and running WSO2 ML, see <u>Getting Started</u>.
- 2. Download and install jq (CLI JSON processor). For instructions, see jq Documentation.
- 3. If you are using Mac OS X, download and install GNU stream editor (sed). For instructions, see <u>GNU sed Documentation</u>.

#### Executing the sample

Follow the steps below to execute the sample.

- 1. Navigate to <ML\_HOME>/samples/default/stacked-autoencoders/ directory using the CLI.
- 2. Execute the following command to execute the sample: ./model-generation.sh

## Output of the sample

Once the sample is successfully executed, you can view the prediction of the model. By default , the sample generates the model in the <ML\_HOME>/models/ directory of your machine. For an example, the generated file is in the following format denoting the date and time when it was generated: wso2-ml-stacked-autoencoders-sample-analysis.Model.2015-11-26\_12-00-46

#### Viewing the model

You can view the summary of the built model using the ML UI as follows.

- 1. Log in to the ML UI from your Web browser using admin/admin credentials and the following URL: https://<ML\_HOST>:<ML\_PORT>/ml
- 2. Click the Projects button as shown below.

Projects 🕥	
Project is a logical grouping of machine learning analyses, which are performed on a dataset. To analyze multiple datasets, you need to create multiple projects.	
You have (1) Projects ADD PROJECT	

3. Click MODELS button of the new analysis which you created by executing the sample as shown below.

Swso2-ml-stacked-autoencoders-sa	Imple-project [created: 2015-11-26 12:00:23:459 ] 1 analysis available	COMPARE MODELS () DELETE PROJECT
This project tests ml workflow for stacked autoencoders	Analysis name	
	e.g. myanalysis (CREATE ANALYSIS	
	vso2-ml-stacked-autoencoders-sample-analysis	VIEW (     MODELS (     DELETE

4. You view the built new model as shown below.



#### Viewing the model prediction

The sample executes the generated model on the

<ML\_HOME>/samples/default/stacked-autoencoders/prediction-test data set, and it prints the value [1.0] as the prediction result In the CLI logs.

# Anomaly Detection Algorithm

- First the dataset will be clustered using K means algorithm according to hyper parameters that user provided.
- In a real world scenario of anomaly detection, positive(anomaly) instances are very rare. Hence, we assume that those anomalies will be outside the clusters.
- So we can detect them by calculating the cluster boundaries. This is how we identify the cluster boundaries,
  - First calculate all the distances between data points and their respective cluster centers.
  - Then select the percentile value from distances of each clusters as their cluster boundaries.
- When a new data point comes, the closest cluster center will be calculated by K means predict function.
- Then the distance between new data point and Its cluster center will be calculated. If it is less than the percentile distance value it is considered as a normal data. If it is greater than the percentile distance value it is considered as an anomaly since it is in outside the cluster.



Steps for Building an Anomaly Detection Model using WSO2 ML

#### Step 1 - Create an Analysis

Upload the dataset and create a new project

As for the every model first you have to upload a dataset and create a new project. Then start a new analysis to build an anomaly detection model

WS ACHINE LEARNER			admin 👤
PROJECTS \ Select a dataset -			
	(+) CREATE PROJECT		
Swso2-ml-anomaly-detection-label	ed-data-sample-project [created: 2015-11-17 18:20:47.783] 1 analysis available	COMPARE MODELS (n) DE	ELETE PROJECT
This project tests mi workflow for k means anomaly detection with labeled data model	Analysis name  e.g. myanalysis CREATE ANALYSIS		
	wso2-ml-anomaly-detection-labeled-data-sample-analysis	Delete	

### Step 2 - Algorithm Selection

In the Algorithm selection process there is a new category called Anomaly Detection. Under that category there are two algorithms. If your dataset is a labeled one you can select K Means Anomaly Detection with Labeled Data. Otherwise you can select K Means Anomaly Detection with Unlabeled Data. There are few model configurations that user have to input in this step.

#### K Means Anomaly Detection with Labeled Data

- Response variable
- Normal label(s) values
- Train data fraction
- Prediction Labels
- Normalization option

₩\$@2 ми	ACHINE LEARNER							admin 👤
PROJECTS \ wso2-	-ml-anomaly-detection-label	ed-data-sample-project \	wso2-ml-anomaly-detection	n-labeled-data-sample-analysis	CANCEL			NEXT ()
		Step 1 Preprocess	<sub>Step 2</sub> Explore	<sub>Step 3</sub> Algorithms	<sub>Step</sub> 4 Parameters	<sub>Step 5</sub> Model		
			Algorithm					
			Algorithm name *					
			K-MEANS WITH LABELE	D DATA		•		
			Response variable *			·		
			Normal label(s) values					
			Train data fraction *					
			0.7					
			Edit text of Prediction L Normal label * normal	abels				
			Anomaly label *					
			anomaly					
			🗑 Normalize data					

#### K Means Anomaly Detection with Unlabeled Data

- Prediction Labels
- Normalization option

WS 2 MACHINE LEARNER						2	admin 👤
PROJECTS \ wso2-ml-anomaly-detection-lab	eled-data-sample-project \ v	vso2-ml-anomaly-detection	n-labeled-data-sample-analysis	(X) CANCEL			NEXT 🕥
	Step 1 Preprocess	<sub>Step 2</sub> Explore	<sub>Step 3</sub> Algorithms	<sub>Step</sub> 4 Parameters	Step 5 Model		
		Algorithm					
		Algorithm name *					
		K-MEANS WITH UNLAE	ELED DATA				
		Edit text of Prediction	Labels				
		Normal label *					
		Anomaly label *					
		anomaly					
		👿 Normalize data					

If there are any categorical features exist on the dataset other than response variable you will be asked to drop them when you proceeds to next step.

Step 1 Preproce	step 2 SS Explore	Step 3 Algorithms	Step 4 Parameters	Step 5 Model	
	All input v	ariables for K-means algo	prithm should be numerica Back to Pre-processing	il.	
	Edit text of Predic	tion Labels			

# Step 3 - Hyper Parameters

In the parameter selection step you have to input necessary hyper parameters for the model

- Maximum Iterations
- Number of Normal Clusters (Since this anomaly detection algorithm have implemented based on K means clustering you have to input the number of normal clusters should build in the model)

WS@2	MACHINE LEARNER						admin 👤
PROJECTS \	vso2-ml-anomaly-detection-lab	eled-data-sample-project \ w	so2-ml-anomaly-detection	i-labeled-data-sample-analysis	(X) CANCEL		PREVIOUS NEXT S
		<sub>Step 1</sub> Preprocess	<sub>Step 2</sub> Explore	<sup>Step 3</sup> Algorithms		Step 5 Model	
			Parameters				
	Set Hyper-Parameters for Anomaly Detection\ K MEANS & LABELED DATA			Anomaly Detection\ K MEANS A	NOMALY DETECTION W	итн	
			Max Iterations 🔞				
			100				
			Num of Normal Clusters	0			
			5				

## Step 4 - Model Building

Then after selecting the dataset version you can build the model.

WS A MACHINE LEARNER						admin 👤
PROJECTS \ wso2-ml-anomaly-detection-labeled	d-data-sample-project \ \	vso2-ml-anomaly-detection	n-labeled-data-sample-analysis	🗙 CANCEL		O PREVIOUS RUN 🕥
	Step 1 Preprocess	<sub>Step 2</sub> Explore	<sub>Step 3</sub> Algorithms	<sub>Step</sub> 4 Parameters	Step 5 Model	
		Model				
		Dataset version				
		IndiansDiabetes-1.0.0			-	

#### Step 5 - Model Summary

After successfully build the model If you had labeled data you can view the model summary. In the summary you can get an overall idea about the build model. There will be very useful information about the model such as F1 score and some other important accuracy measures, confusion matrix, cluster diagram etc. So based on these information you will be able pick a better model.

Model is evaluated for range of percentile values that means for rage of cluster boundaries to pick the best one. There for in the model summary by default you will see the measures with respect to best percentile value. But you can see how measures are changing according to the percentile by moving the percentile slider. Based on that you can get an idea about the best percentile value to use for predictions. By default we use the percentile range as 80 - 100. But if you need a different range to evaluate the model you can change the range by input minPercentile and maxPercentile as system properties once you start the server. Keep in mind that you need to input values between 0 - 100 as percentiles. You can input system properties when you starting the server as below.

./wso2server.sh -DminPercentile=60 -DmaxPercentile=90





#### Step 6 - Prediction

This is where you can predict new data using built model. As a input you have to give feature values of new data point or you can give new data as a batch using csv or tsv file. Other than the data you should input the percentile value to identify the cluster boundaries. Default value is there. You can just keep it if you don't have a clear idea about that. If you had labeled data when building the model it will set the optimum value as the default value which obtained from the model evaluation. So after input those values you will get the predictions for new data.

WS MACHINE LEARNER		admin <u>1</u>
PROJECTS \ wso2-ml-anomaly-detection-lab data-sample-analysis.Model.2015-11-17_	eled-data-sample-project \ wso2-ml-anomaly-detection-labeled-data-sample-analysis \ wso2-ml-anomaly-detection-labeled- 18-26-51	
	Predict	
	Prediction Source  File	
	Data File (max size: 100MB) * Browse wdbc.csv	
	Data Format  CSV	
	Column Header Available • Yes	
	Percentile Value * 96	
	Predict	

# Anomaly Detection for WSO2 ML - Samples

Generating a Model Using the K Means Anomaly Detection Algorithm with unlabeled data

- Introduction
- Prerequisites
- Executing the sample
- Output of the sample

#### Introduction

This sample demonstrates how a model is generated out of a unlabeled data set using the k-means anomaly detection algorithm. The sample uses a data set to generate a model.

#### **Prerequisites**

Follow the steps below to set up the prerequisites before you start.

- 1. Download WSO2 Machine Learner, and start the server. For information on setting up and running WSO2 ML, see <u>Getting Started</u>.
- 2. Download and install jq (CLI JSON processor). For instructions, see jq Documentation.
- 3. If you are using Mac OS X, download and install GNU stream editor (sed). For instructions, see <u>GNU sed Documentation</u>.

#### **Executing the sample**

Follow the steps below to execute the sample.

- 1. Navigate to <ML\_HOME>/samples/default/anomaly-detection-unlabeled-data/ directory using the CLI.
- 2. Execute the following command to execute the sample: ./model-generation.sh

#### Output of the sample

Once the sample is successfully executed, you can view the prediction of the model as described below.

lcon

By default, the sample generates the model in the <ML\_HOME>/models/ directory of your machine. For example, the generated file is in the following format denoting the date and time when it was generated:

wso2-ml-anomaly-detection-unlabeled-data-sample-analysis.Model.2015-11-13\_10-51-27

#### Viewing the model

You can view the summary of the built model using the ML UI as follows.

- 1. Log in to the ML UI from your Web browser using admin/admin credentials and the following URL: https://<ML\_HOST>:<ML\_PORT>/ml
- 2. Click the Projects button as shown below.



3. Click MODELS button of the new analysis which you created by executing the sample as shown below.

Swso2-ml-anomaly-detection-unl			COMPARE MODELS (n) DELETE PROJECT
This project tests ml workflow for k means anomaly detection with unlabeled data model	Analysis name 💿		
	e.g. myanalysis	CREATE ANALYSIS	
	so2-ml-anomaly-detection-unlabeled-data-sample-analysis		(A) VIEW (A) MODELS (A) DELETE

4. You view the built new model as shown below.

wso2-ml-anomaly-detection-unlabeled-data-sample-analysis.Model.2015-11-13_10-51-27 [Created: 2015-11-13 10:51:27485] [Dataset version: seeds-1.0.0] [Status: Complete 2]	🕞 PREDICT 🔔 DOWNLOAD 🔕 PUBLISH
	(n) DELETE MODEL

#### Viewing the model prediction

The sample executes the generated model on the <ML\_HOME>/samples/default/k-means-with-unlabeled-data/prediction-test data set, and it prints
the value ["anomaly"] as the prediction result In the CLI logs.

# Generating a Model Using the K Means Anomaly Detection Algorithm with labeled data

- Introduction
- Prerequisites

- Executing the sample
- Output of the sample

#### Introduction

This sample demonstrates how a model is generated out of a labeled data set using the k-means anomaly detection algorithm. The sample uses a data set to generate a model, which is divided into two sets for training and testing.

#### Prerequisites

Follow the steps below to set up the prerequisites before you start.

- 1. Download WSO2 Machine Learner, and start the server. For information on setting up and running WSO2 ML, see <u>Getting Started</u>.
- 2. Download and install jq (CLI JSON processor). For instructions, see jq Documentation.
- 3. If you are using Mac OS X, download and install GNU stream editor (sed). For instructions, see <u>GNU sed Documentation</u>.

#### Executing the sample

Follow the steps below to execute the sample.

- 1. Navigate to <ML\_HOME>/samples/default/anomaly-detection-labeled-data/ directory using the CLI.
- 2. Execute the following command to execute the sample: ./model-generation.sh

#### Output of the sample

Once the sample is successfully executed, you can view the summary and the prediction of the model as described below.

Icon

By default, the sample generates the model in the <ML\_HOME>/models/ directory of your machine. For example, the generated file is in the following format denoting the date and time when it was generated:

```
wso2-ml-anomaly-detection-labeled-data-sample-analysis.Model.2015-11-13_10-56-15
```

#### Viewing the model

You can view the summary of the built model using the ML UI as follows.

- 1. Log in to the ML UI from your Web browser using admin/admin credentials and the following URL: https://<ML\_HOST>:<ML\_PORT>/ml
- 2. Click the Projects button as shown below.

Projects 🕥	
Project is a logical grouping of machine learning analyses, which are performed on a dataset. To analyze multiple datasets, you need to create multiple projects.	
You have (1) Projects ADD PROJECT	

3. Click MODELS button of the new analysis which you created by executing the sample as shown below.

$\odot$ wso2-ml-anomaly-detection-labele	d-data-sample-project			COMPARE MODELS	DELETE PROJECT
This project tests ml workflow for k means anomaly	Analysis name 🔞				
detection with labeled data model	e.g. myanalysis	CREATE ANALYSIS			
	wso2-ml-anomaly-detection	on-labeled-data-sample-analysis	(	VIEW 🚯 MODELS 💼 DELETE	
	wso2-ml-anomaly-detection	on-labeled-data-sample-analysis	([	) view 🚯 models 💼 delete	

4. Click VIEW of the built new model as shown below.



You view the summary of the built model as shown below.

leasures Measure Value (%) Precision 66.18 Recall 96.48 Accuracy 65.91	Cluster Diagram	m	anon norr	naly nal		ar - -	nomaly 137 70		•		rmal 5 B	
leasures Measure Value (%) Precision 66.18 Recall 96.48 Accuracy 65.91	Cluster Diagram	m	anon norr	naly nal		•	137 70 .		•		8	
leasures           Measure         Value (%)           Precision         66.18           Recall         96.48           Accuracy         65.91	Cluster Diagram	m	norr	nal		:	70		•		8	
Measure     Value (%)       Precision     66.18       Recall     96.48       Accuracy     65.91	Cluster Diagram	m	. :	:	•		•		•			
MeasureValue (%)Precision66.18Recall96.48Accuracy65.91	- 180 - 160 - 140 - 120		• •	:	•	•	•		•			
Precision     66.18       Recall     96.48       Accuracy     65.91	160 - 140 - 120 -		:	:	;							
Recall 96.48 Accuracy 65.91	140 -	: :	: 1									
Accuracy 65.91	120 -		· 1			-			:		•	
				1		-	:	:				
	80-			:	1	2	:	•			•	
	60 -						•		•			
	40 -											
	20 -											
	0.0 1.0 - 2 +	2.0 3.	8.0 4.0	5.0	6.0	7.0	8.0	9.0	10	11 Numi	12 Pregnan	cies
	X-Axis				Y	-Axis	D					

#### Model Summary [F1 Score: 0.7851]

#### Viewing the model prediction

The sample executes the generated model on the

<ML\_HOME>/samples/default/k-means-with-labeled-data/prediction-test data set, and it prints the value ["anomaly"] as the prediction result In the CLI logs.

idmin 👤

# Generating a Tuned Model Using the K Means Anomaly Detection Algorithm with labeled data

- Introduction
- Prerequisites
- Executing the sample
- Output of the sample

#### Introduction

This sample demonstrates how a model is generated out of a dataset using the k-means anomaly detection algorithm using <u>tuned hyper parameter values</u>. You can find these parameter values in the <ML\_HOME>/samples/tuned/k-means-with-labeled-data/hyper-parameters file. The sample uses a data set to generate a model, which is divided into two sets for training and testing.

#### Prerequisites

Follow the steps below to set up the prerequisites before you start.

- 1. Download WSO2 Machine Learner, and start the server. For information on setting up and running WSO2 ML, see <u>Getting Started</u>.
- 2. Download and install jq (CLI JSON processor). For instructions, see jq Documentation.
- 3. If you are using Mac OS X, download and install GNU stream editor (sed). For instructions, see <u>GNU sed Documentation</u>.

#### Executing the sample

Follow the steps below to execute the sample.

- 1. Navigate to <ML\_HOME>/samples/tuned/anomaly-detection-labeled-data/ directory using the CLI.
- 2. Execute the following command to execute the sample: ./model-generation.sh

#### Output of the sample

Once the sample is successfully executed, you can view the summary and the prediction of the model as described below.

Icon

By default, the sample generates the model in the <ML\_HOME>/models/ directory of your machine. For example, the generated file is in the following format denoting the date and time when it was generated:

wso2-ml-anomaly-detection-labeled-data-tuned-sample-analysis.Model.2015-11-13\_11-00-21

#### Viewing the model summary

You can view the summary of the built model using the ML UI as follows.

- Log in to the ML UI from your Web browser using admin/admin credentials and the following URL: https://<ML\_HOST>:<ML\_PORT>/ml
- 2. Click the Projects button as shown below.



3. Click MODELS button of the new analysis which you created by executing the sample as shown below.

🛇 wso2-ml-anomaly-detection-label	ed-data-tuned-sample-project [created: 2015-11-13 10:59:58.813 ] 1 analysis available	COMPARE MODELS () DELETE PROJECT
This project tests ml workflow for k means anomaly detection with labeled data model with tuned hyper- parameters	Analysis name • e.g. myanalysis • CREATE ANALYSIS	
	so2-ml-anomaly-detection-labeled-data-tuned-sample-analysis	(a) VIEW (b) MODELS (a) DELETE

4. Click VIEW of the built new model as shown below.



You view the summary of the built model as shown below.



#### Viewing the model prediction

The sample executes the generated model on the

<ML\_HOME>/samples/tuned/k-means-with-labeled-data/prediction-test data set, and it prints the value ["positive"] as the prediction result In the CLI logs.

# **PMML Support**

# PMML

The Predictive Model Markup Language (PMML) is an XML-based file format developed by the Data Mining Group to provide a way for applications to describe and exchange models produced by data mining and machine learning algorithms. It supports common models such as logistic regression and feedforward neural networks.

Since PMML is an XML-based standard, the specification comes in the form of an XML schema.

# PMML Support in WSO2 ML

WSO2 Machine Learner supports PMML export and publish functionality. Prior to this WSO2 ML was able to generate the serialized model only. From this release onwards a user could use that serialized model to export(download) or publish into PMML format. In order to generate the PMML model the user should already have the serialized model generated within the Machine Learner. If the user already has the serialized model, by passing that specific model's modelld into the export or publish APIs, the PMML model could be generated.



# Usage

PMML export feature is used to export a serialized model created by WSO2 Machine Learner to PMML format. Currently the following model types can be exported as PMML.

- Linear Regression
- Logistic Regression
- Ridge Regression
- Lasso Regression
- SVM

• K-Means

PMML export for other model types can be expected in future releases. Apart from being able to export those models users can also publish them to the WSO2 registry.

Downloading a model in PMML Format

Method 1

After creating an analysis and generating a model click on "MODELS" button. The resulting page will be as follows. Click on "DOWNLOAD" button.

wso2-ml-linear-regression-sample-analysis.Model.2015-11-16\_14-47-14 [Greated 2015-11-16/14-47.14.007]

In the resulting dialog box choose "PMML" as download type.

Please select download type	
Serialized PMML	

#### Method 2

After creating an analysis and generating a model click on "COMPARE MODELS" button. The resulting page will be as follows. Click on "DOWNLOAD" button.

Model Comparison		
Classification Numerical Prediction Clustering		
wso2-ml-linear-regression-sample-analysis.Model.2015-11-16_14-47-14 [Algorithm: LINEAR REGRESSION] [Mean squared error: 1.014+2]		
Show 10 •		1

In the resulting dialog box choose "PMML" as download type.

Please select download type	
Serialized PMML	

### Publishing a model to Registry in PMML Format

#### Method 1

After creating an analysis and generating a model click on "MODELS" button. The resulting page will be as follows. Click on "PUBLISH" button.

wso2-ml-linear-regression-sample-analysis.Model.2015-11-16_14-47-14	
[Dataset version: abalone-linear reĝression-dataset-1.0.0] [Status: Complete ♥]	

In the resulting dialog box choose "PMML" as publish type.



#### Method 2

After creating an analysis and generating a model click on "COMPARE MODELS" button. The resulting page will be as follows. Click on "PUBLISH" button.

Model Comparison	
Classification Numerical Prediction Clustering	
wso2-ml-linear-regression-sample-analysis.Model.2015-11-16_14-47-14 [Agonthm: LINEAR REGRESSION] [Mean squared error: 1.01e+2]	
Show 10 •	1

#### In the resulting dialog box choose "PMML" as publish type.

Please select model type to publish	
Sertalized PMMIL	

# APIs

Finding whether an algorithm supports an export type

Overview

Description	Retrieve information whether a given algorithm supports an export type
Resource Path	<pre>/api/configs/algorithms/{algorithmName}/expo rtable?format={exportFormat}</pre>
HTTP Method	GET
Request/Response Format	application/json

Parameter description

Parameter	Description
{algorithmName}	Name of the algorithm which needs to be queried on a specific export type
{exportFormat}	The export format

Sample cURL command

```
curl -H "Content-Type: application/json" -H "Authorization: Basic
YWRtaW46YWRtaW4="
https://localhost:9443/api/configs/algorithms/LINEAR_REGRESSION/expo
rtable?format=pmml -v -k
```

#### Example

#### GET

https://localhost:9443/api/configs/algorithms/{algorithmName}/?format={ex
portFormat}

Sample output

HTTP/1.1 200 OK

REST API response

HTTP status	200, 404 or 400.
code	
	For descriptions of the HTTP status codes, see HTTP Status
	Codes.

# Export a model

#### Overview

Description	Export a serialized model in either PMML or serialized formats
Resource Path	<pre>api/models/{modelId}/export?mode={exportType }</pre>
HTTP Method	GET
Request/Response Format	application/json

# Parameter description

Parameter	Description
{modelId}	model ID of the model to be exported
{exportType}	The export type

Sample cURL command

```
curl -H "Content-Type: application/json" -H "Authorization: Basic
YWRtaW46YWRtaW4=" -v
https://localhost:9443/api/models/1/export?mode=pmml -k
```

#### Example

#### GET

```
https://localhost:9443/api/models/{modelId}/export?mode={exportFormat}
```

#### Sample output

```
<?xml version="1.0" encoding="UTF-8"?><PMML xmlns="http://www.dmg.org/PMML-4_2"
version="4.2">
   <Header description="linear regression">
       <Application name="Apache Spark MLlib"/>
        <Timestamp>2015-11-18T18:02:50</Timestamp>
   </Header>
    <DataDictionary numberOfFields="9">
        <DataField dataType="double" name="field_0" optype="continuous"/>
       <DataField dataType="double" name="field_1" optype="continuous"/>
       <DataField dataType="double" name="field_2" optype="continuous"/>
        <DataField dataType="double" name="field 3" optype="continuous"/>
       <DataField dataType="double" name="field_4" optype="continuous"/>
       <DataField dataType="double" name="field 5" optype="continuous"/>
       <DataField dataType="double" name="field_6" optype="continuous"/>
       <DataField dataType="double" name="field_7" optype="continuous"/>
        <DataField dataType="double" name="target" optype="continuous"/>
    </DataDictionary>
    <RegressionModel functionName="regression" modelName="linear regression">
        <MiningSchema>
            <MiningField name="field_0" usageType="active"/>
            <MiningField name="field_1" usageType="active"/>
            <MiningField name="field_2" usageType="active"/>
            <MiningField name="field 3" usageType="active"/>
            <MiningField name="field_4" usageType="active"/>
            <MiningField name="field 5" usageType="active"/>
            <MiningField name="field_6" usageType="active"/>
            <MiningField name="field_7" usageType="active"/>
            <MiningField name="target" usageType="target"/>
```

```
</MiningSchema>

<RegressionTable intercept="0.0">

<NumericPredictor coefficient="0.18637180966084266" name="field_0"/>

<NumericPredictor coefficient="0.09860314154841378" name="field_1"/>

<NumericPredictor coefficient="0.07703949089962057" name="field_2"/>

<NumericPredictor coefficient="0.02658539142216247" name="field_3"/>

<NumericPredictor coefficient="0.1661298483263787" name="field_4"/>

<NumericPredictor coefficient="0.0707509545901941" name="field_5"/>

<NumericPredictor coefficient="0.03607354242895441" name="field_6"/>

<NumericPredictor coefficient="0.04851007262677507" name="field_7"/>

</RegressionTable>

</PMML>
```

#### **REST API response**

HTTP status	200, 400, 404 or 500.
Coue	For descriptions of the HTTP status codes, see HTTP Status Codes.

#### Publish a model

#### Overview

Description	Publish a model
Resource Path	api/models/{modelId}/publish?mode={publishTy pe}
HTTP Method	POST

Request/Response	application/json
Format	

Parameter description

Parameter	Description
{modelld}	model ID of the model to be published
{publishType}	The model type to be published

Sample cURL command

curl -X POST -H "Content-Type: application/json" -H "Authorization: Basic YWRtaW46YWRtaW4=" -v https://localhost:9443/api/models/1/publish?mode=pmml -k -v

Example

#### GET

https://localhost:9443/api/models/{modelId}/publish?mode={publishType}

Sample output

HTTP/1.1 200 OK

**REST API response** 

HTTP status code	200, 400, 404 or 500.
	For descriptions of the HTTP status codes, see HTTP Status Codes.

# Samples

Each generic sample which supports PMML export is extended with exporting the model to PMML format. Once the PMML model is generated it will be printed in the client's console.

e.g

{ML\_HOME}/samples/default/linear-regression/.model-generation.sh