

K-means with MLlib

Use the MLlib implementation of K-means Clustering to cluster your data



Product: IBM® SPSS® Modeler

Extension type: Analysis

Table of Contents

Description.....	3
Requirements.....	3
Installation.....	3
Python Packages used.....	3
User Interface.....	4-5
Example.....	6-10
Important links.....	10
Learn.....	10
Discuss.....	10
References.....	11

**Description:**

K-means clustering is a very popular algorithm used for clustering data. This extension uses the PySpark MLlib implementation of this algorithm. In order to run K-means clustering, you need to specify the number of clusters you want. This can be determined using domain knowledge about your dataset or through trial and error of evaluating different cluster parameters. Additional information on the parameters used by this algorithm will be discussed in the user interface portion of the documentation.

Requirements:

- SPSS Modeler v18.0 or later
- [Python 2.7 Anaconda distribution](#)

Installation:Initial one-time set-up for PySpark Extensions

If using v18.0 of SPSS Modeler, navigate to the options.cfg file (Windows default path: C:\Program Files\IBM\SPSS\Modeler\18.0\config). Open this file in a text editor and paste the following text at the bottom of the document:

```
eas_pyspark_python_path, "C:/Users/IBM_ADMIN/Anaconda/python.exe"
```

The underlined path should be replaced with the path to your python.exe from your Anaconda installation.

Extension Hub Installation

1. Go to the Extension menu Modeler and click "Extension Hub"
2. In the search bar, type the name of this extension and press enter
3. Check the box next to "Get extension" and click OK at the bottom of the screen
4. The extension will install and a pop-up will show what palette it was installed

Manual Installation

1. Save the .mpe file to your computer
2. In Modeler, click the Extensions menu, then click Install Local Extension Bundle
3. Navigate to where the .mpe was saved and click open
4. The extension will install and a pop-up will show what palette it was installed

Python Libraries used:

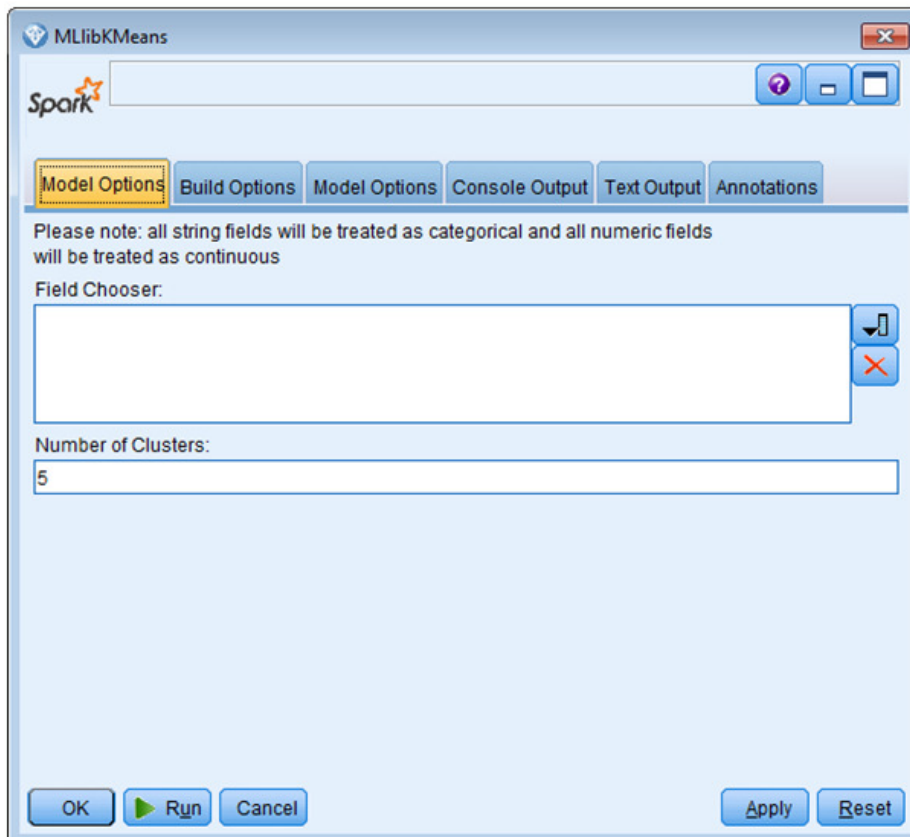
- Spark MLlib – [K-means Clustering](#)

User Interface

This extension has two tabs which allow you to control the features used by the model as well as additional parameters that can be tuned to optimize performance.

Tab 1 – Model Options

- Field Chooser – Select the fields from your data you could like to include for clustering. A note in the dialog reminds that all string fields will be treated as categorical and numeric will be treated as continuous. If you have a number that functions as a category (e.g. 1/0), then derive a string version of that field for use in the clustering.
- Number of Clusters – As mentioned in the description, the number of desired clusters must be explicitly stated prior to the execution of the algorithm.



MLibKMeans

Spark

Model Options Build Options Model Options Console Output Text Output Annotations

Please note: all string fields will be treated as categorical and all numeric fields will be treated as continuous

Field Chooser:

Number of Clusters:

5

OK Run Cancel Apply Reset



Tab 2 – Build Options

For more details on the parameters of this function [visit the MLlib documentation](#)

- Epsilon - distance threshold to determine convergence of k-means
- Number of iterations - maximum number of iterations to run clustering
- Number of Runs - the number of times to run the k-means algorithm
- Initialization Steps – used with K-Means || algorithm, determines the number of steps
- Initialization Mode - either random initialization or initialization via k-means||
- Random Seed – this value will be used if random is selected for the initialization mode

MLlibKMeans

Spark

Model Options **Build Options** Model Options Console Output Text Output Annotations

Epsilon (Convergence Threshold): 0.0001

Number of Iterations: 100

Number of Runs: 1

☒ Use Random Seed

Initialization Steps: 5

Initialization Mode

☒ Random

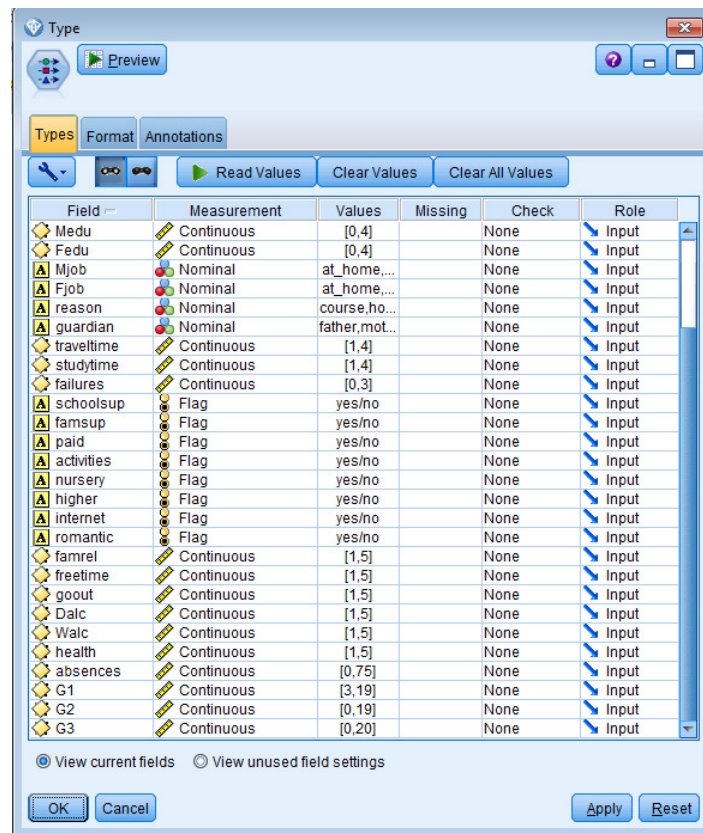
☐ K-Means||

Random Seed: 0

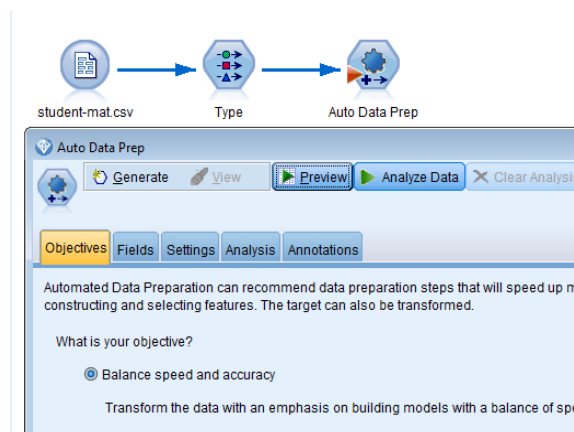
OK Run Cancel Apply Reset



2. Next we can add a Type node, to read values to make sure all the meta-data is correct.

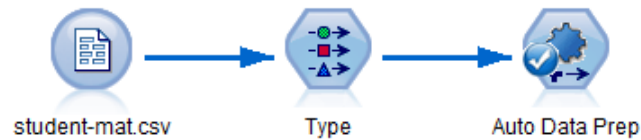


3. Before we cluster our data, we should do some pre-processing to make sure our continuous variables are on the same scale. This is easily done with the Auto Data Prep node in Modeler
 - a. Add the Auto Data Prep node from the Field Ops palette
 - b. Open the node by double clicking and click the button to “Analyze Data”





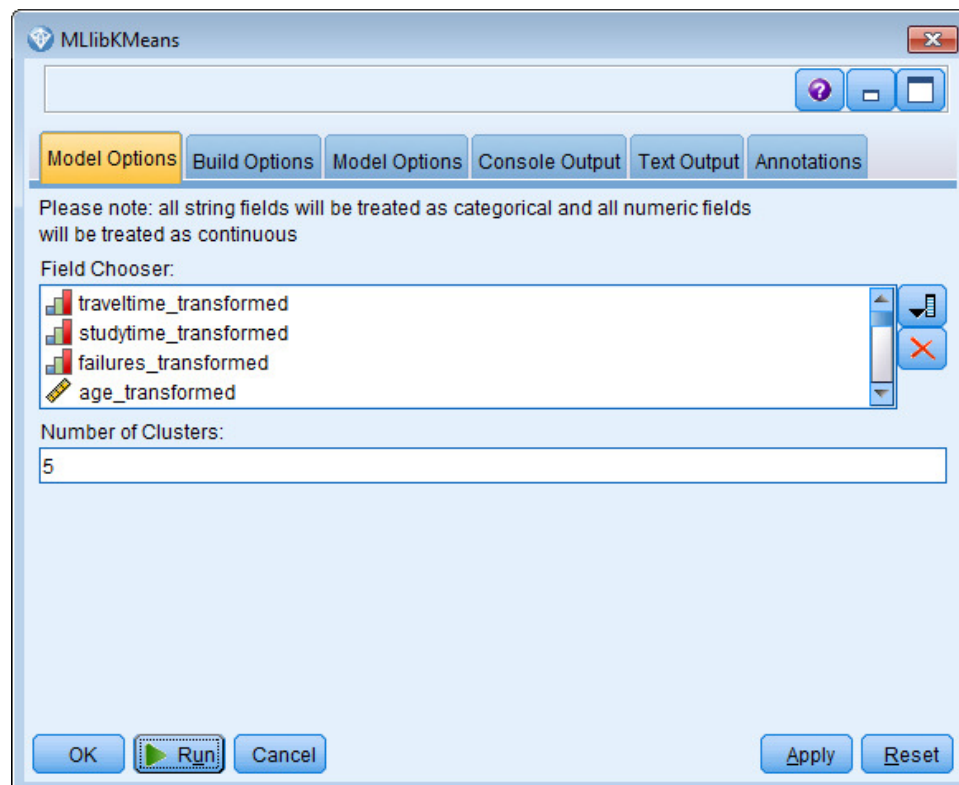
- c. This will create dummy variables for categorical variables and will standardize numeric variables
- d. You will know this step is complete because a check mark will appear on the node



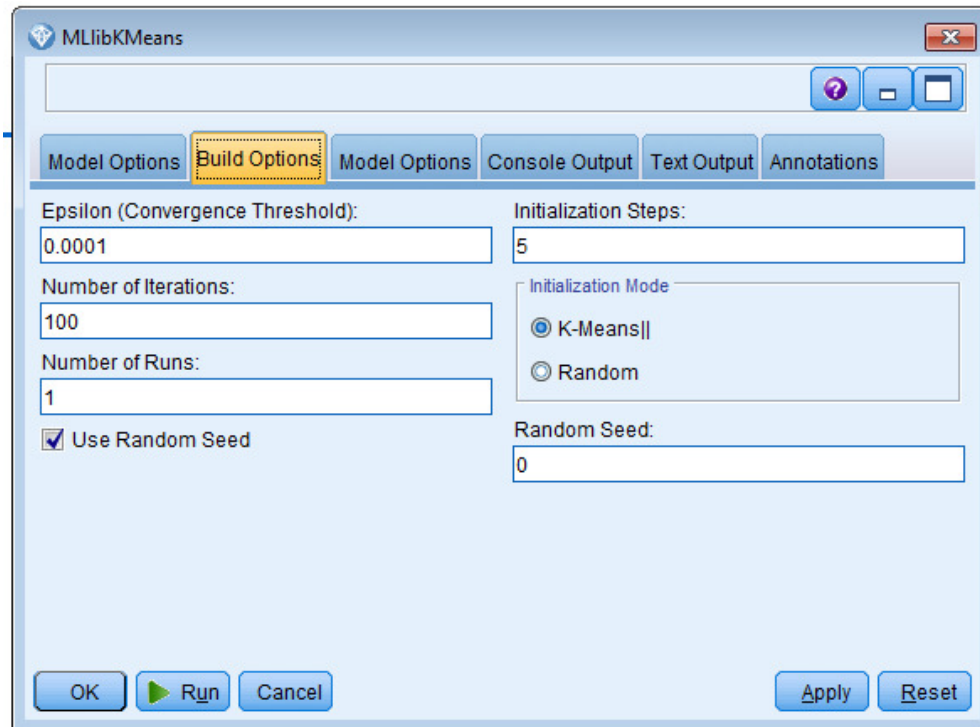
4. Now, let's add our K-Means node so our stream will look like this



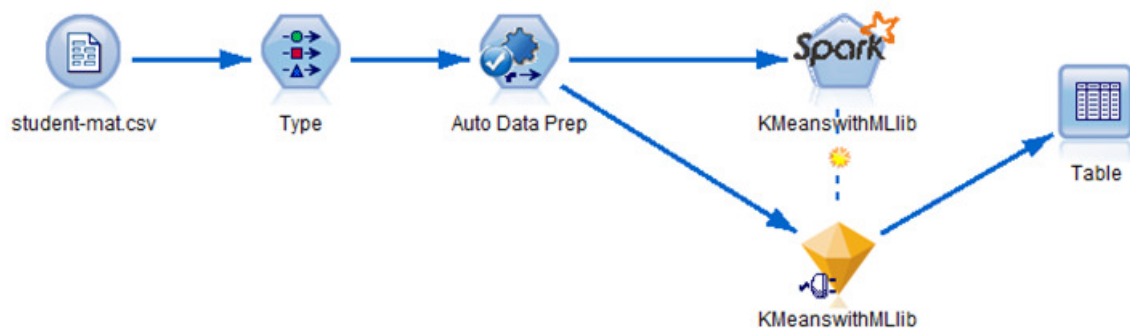
5. Open the MLlib K-Means dialog
- a. Add all the Fields
 - b. Since we do not have more knowledge of the dataset , let's use the default 5 clusters



- c. Let's also use the default options for the algorithm; these can be changed on the Build Options tab.



6. Run this stream to create the model.
7. Now add a table node connected to the model nugget from the Output palette
8. Right click the table node and run to get the following stream:





9. The table produced will include the cluster assignment for each student:

Table (34 fields, 395 records)

File Edit Generate

Table Annotations

	amrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3	\$K-cluster
1	4	3	4	1	1	3	6	5	6	6	4
2	4	3	2	1	2	5	4	6	10	10	0
3	4	2	2	1	1	1	0	16	18	19	1
4	4	3	3	1	3	5	2	14	14	14	1
5	3	2	3	1	2	2	6	13	14	14	1
6	4	4	1	1	1	1	0	13	14	15	1
7	4	3	2	1	1	5	2	10	9	8	0
8	5	3	3	1	1	5	4	11	11	11	0
9	4	5	2	1	1	5	0	17	16	16	1
10	5	4	3	1	1	4	2	15	16	18	1
11	3	3	3	1	2	3	25	7	10	11	2
12	4	3	3	2	2	5	14	10	10	9	2
13	4	3	3	2	2	5	2	15	15	14	1
14	5	5	5	3	4	5	6	11	11	10	0
15	4	3	2	1	1	1	0	14	15	15	1
16	2	4	4	2	3	4	6	10	11	11	0
17	4	4	4	2	4	2	0	10	10	10	0
18	4	1	3	1	3	4	2	8	9	8	0
19	3	3	4	2	4	5	2	8	6	5	4
20	3	4	3	1	1	1	8	11	11	10	0

OK

This clustering assignment can now be used for other steps in our Modeler workflow.

Important Links

Learn

- Learn more about [SPSS software](#).
- To learn more about this implementation of K-means take a look at the [Spark documentation](#)
- Read about coding a [PySpark Extension for SPSS Modeler](#)
- Visit [developerWorks Business analytics](#) for more technical analytics resources for developers.

Discuss

- Visit the [IBM SPSS Community](#) to share tips and experiences with other IBM SPSS developers.
- Follow [developerWorks on Twitter](#) to be among the first to hear about new resources.

References

- [1] Apache Spark version 1.6.1 – Mllib Guide: K-means (<http://spark.apache.org/docs/latest/mllib-clustering.html#k-means>)
- [2] Lichman, M. (2013). UCI Machine Learning Repository [<http://archive.ics.uci.edu/ml>]. Irvine, CA: University of California, School of Information and Computer Science.
- [3] P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th FUTURE BUSINESS TECHNOLOGY Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7.