MAHOUT - CLASSIFICATION

What is Classification?

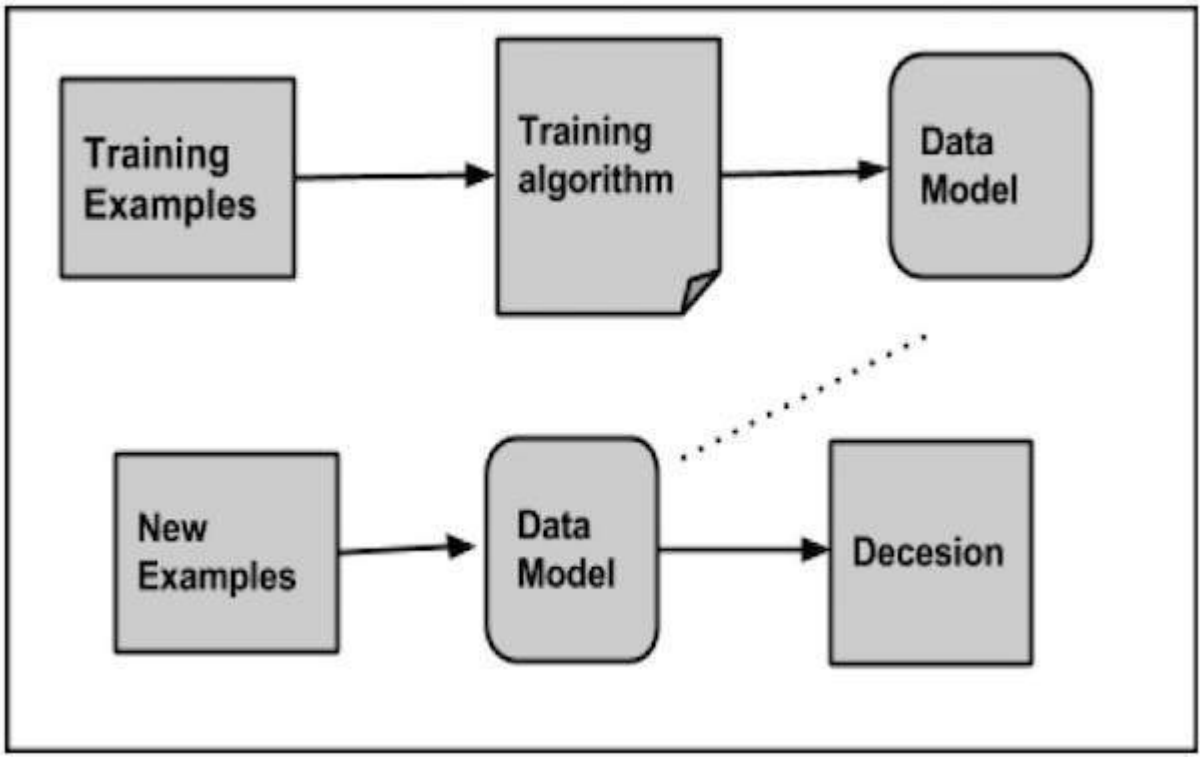
Classification is a machine learning technique that uses known data to determine how the new data should be classified into a set of existing categories. For example,

- iTunes application uses classification to prepare playlists.
- Mail service providers such as Yahoo! and Gmail use this technique to decide whether a new mail should be classified as a spam. The categorization algorithm trains itself by analyzing user habits of marking certain mails as spams. Based on that, the classifier decides whether a future mail should be deposited in your inbox or in the spams folder.

How Classification Works

While classifying a given set of data, the classifier system performs the following actions:

- Initially a new data model is prepared using any of the learning algorithms.
- Then the prepared data model is tested.
- Thereafter, this data model is used to evaluate the new data and to determine its class.



Applications of Classification

- **Credit card fraud detection** - The Classification mechanism is used to predict credit card frauds. Using historical information of previous frauds, the classifier can predict which future transactions may turn into frauds.
- **Spam e-mails** - Depending on the characteristics of previous spam mails, the classifier determines whether a newly encountered e-mail should be sent to the spam folder.

Naive Bayes Classifier

Mahout uses the Naive Bayes classifier algorithm. It uses two implementations:

- Distributed Naive Bayes classification

- Complementary Naive Bayes classification

Naive Bayes is a simple technique for constructing classifiers. It is not a single algorithm for training such classifiers, but a family of algorithms. A Bayes classifier constructs models to classify problem instances. These classifications are made using the available data.

An advantage of naive Bayes is that it only requires a small amount of training data to estimate the parameters necessary for classification.

For some types of probability models, naive Bayes classifiers can be trained very efficiently in a supervised learning setting.

Despite its oversimplified assumptions, naive Bayes classifiers have worked quite well in many complex real-world situations.

Procedure of Classification

The following steps are to be followed to implement Classification:

- Generate example data
- Create sequence files from data
- Convert sequence files to vectors
- Train the vectors
- Test the vectors

Step1: Generate Example Data

Generate or download the data to be classified. For example, you can get the **20 newsgroups** example data from the following link: http://people.csail.mit.edu/jrennie/20Newsgroups/20news-bydate.tar.gz

Create a directory for storing input data. Download the example as shown below.

```
$ mkdir classification_example
$ cd classification_example
$tar xzvf 20news-bydate.tar.gz
wget http://people.csail.mit.edu/jrennie/20Newsgroups/20news-bydate.tar.gz
```

Step 2: Create Sequence Files

Create sequence file from the example using **seqdirectory** utility. The syntax to generate sequence is given below:

```
mahout seqdirectory -i <input file path> -o <output directory>
```

Step 3: Convert Sequence Files to Vectors

Create vector files from sequence files using **seq2parse** utility. The options of **seq2parse** utility are given below:

```
$MAHOUT_HOME/bin/mahout seq2sparse
--analyzerName (-a) analyzerName  The class name of the analyzer
--chunkSize (-chunk) chunkSize    The chunkSize in MegaBytes.
--output (-o) output               The directory pathname for o/p
--input (-i) input                 Path to job input directory.
```

Step 4: Train the Vectors

Train the generated vectors using the **trainnb** utility. The options to use **trainnb** utility are given below:

```
mahout trainnb
 -i ${PATH_TO_TFIDF_VECTORS}
 -el
 -o ${PATH_TO_MODEL}/model
 -li ${PATH_TO_MODEL}/labelindex
 -ow
 -c
```

Step 5: Test the Vectors

Test the vectors using **testnb** utility. The options to use **testnb** utility are given below:

```
mahout testnb
 -i ${PATH_TO_TFIDF_TEST_VECTORS}
 -m ${PATH_TO_MODEL}/model
 -l ${PATH_TO_MODEL}/labelindex
 -ow
 -o ${PATH_TO_OUTPUT}
 -c
 -seq
```