Vectors and Labeled Points

Lesson Objectives

- After completing this lesson, you should be able to:
  - Understand local vectors and labeled points
  - Create dense and sparse vectors
  - Create labeled points

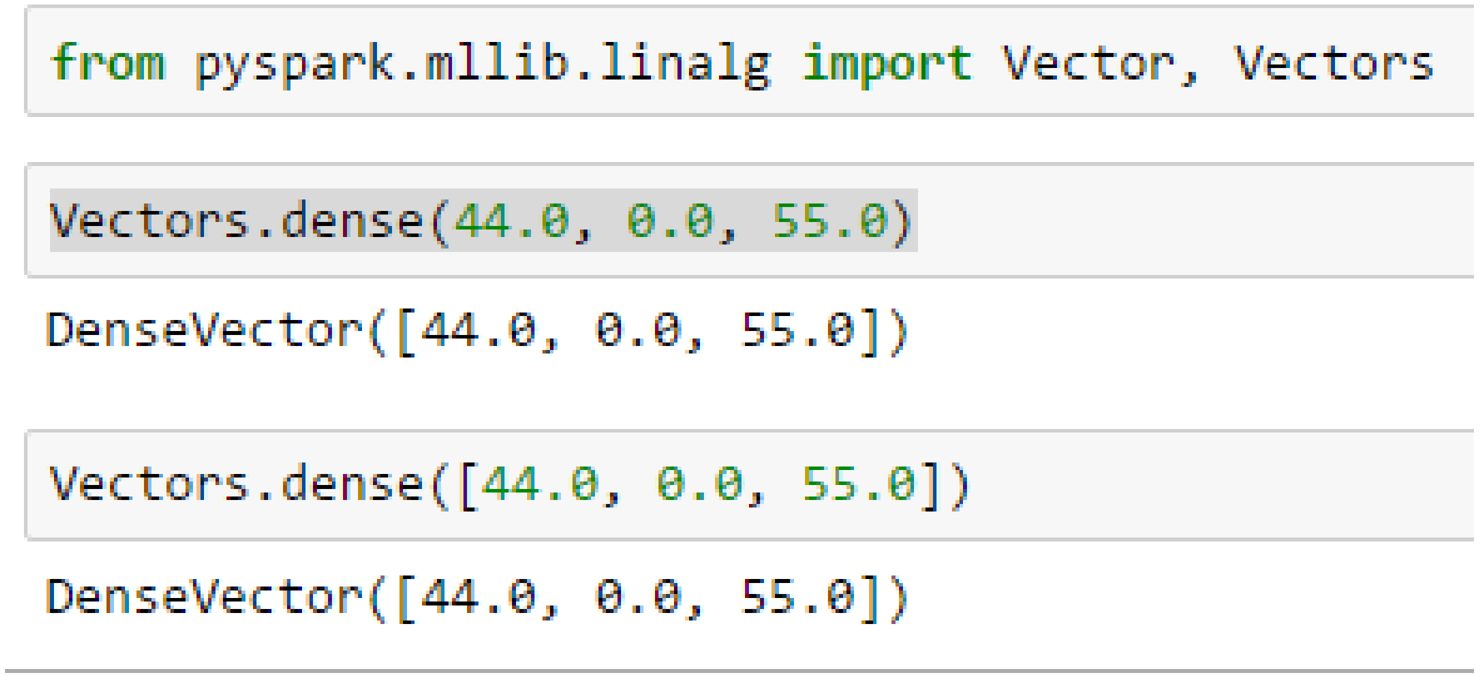
MLlib's Local Vectors

- Linear algebra operations provided by Breeze and jblas
- You have to import `from pyspark.mllib.linalg import Vector, Vectors` to use MLlib Vectors
- Indices are 0-based integers and values are doubles
- Local vectors are stored on a single machine
- MLlib's vectors can be either dense or sparse

Dense Vector

- A dense vector is backed by a double array containing its values
- It is the "obvious" implementation of a Vector
- Easily created from an array of doubles

Dense Vector Example



```
from pyspark.mllib.linalg import Vector, Vectors
```

```
Vectors.dense(44.0, 0.0, 55.0)
```

```
DenseVector([44.0, 0.0, 55.0])
```

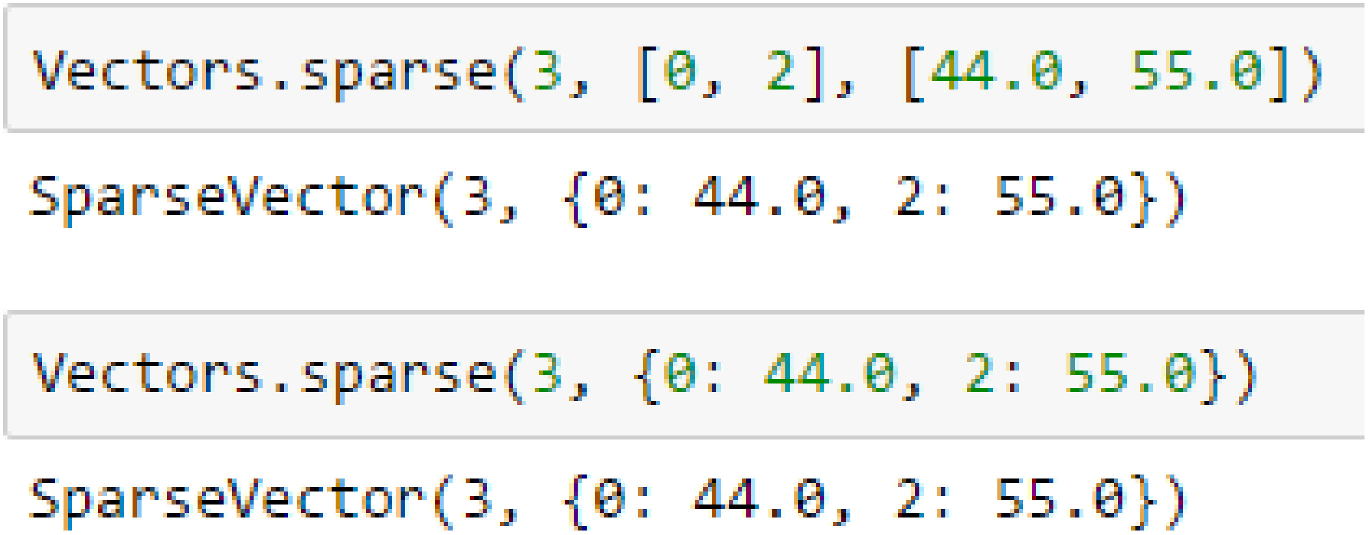
```
Vectors.dense([44.0, 0.0, 55.0])
```

```
DenseVector([44.0, 0.0, 55.0])
```

Sparse Vector

- A sparse vector is backed by two arrays:
  - an integer array representing the indexes
  - a double array containing the corresponding values
- It is a binary-search implementation of a Vector
- Can be created by specifying the indices and values for non-zero entries as either
  - two separate arrays
  - a sequence of tuples

Sparse Vector Examples



```
Vectors.sparse(3, [0, 2], [44.0, 55.0])
```

```
SparseVector(3, {0: 44.0, 2: 55.0})
```

```
Vectors.sparse(3, {0: 44.0, 2: 55.0})
```

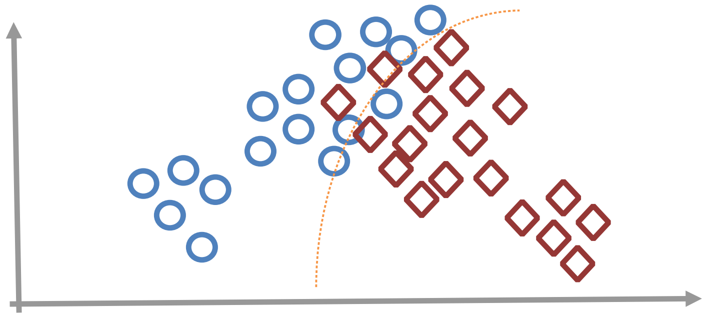
```
SparseVector(3, {0: 44.0, 2: 55.0})
```

Labeled points

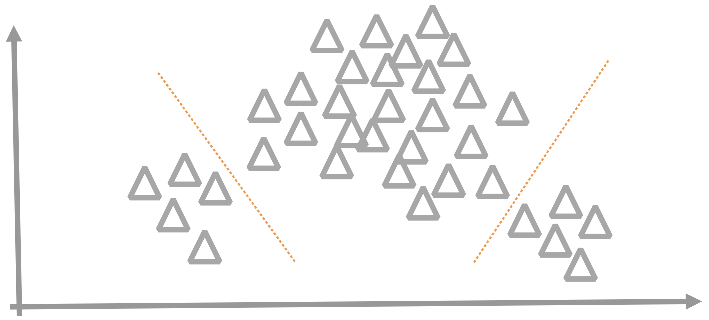
- … are the association of a vector, either dense or sparse, with a corresponding label/response
- … are used in supervised machine learning algorithms

Supervised Learning Algorithms

- •Supervised learning: a machine is told the “correct” answers so it can look for similar patterns



- •Unsupervised learning: where the machine has to make intelligent guesses



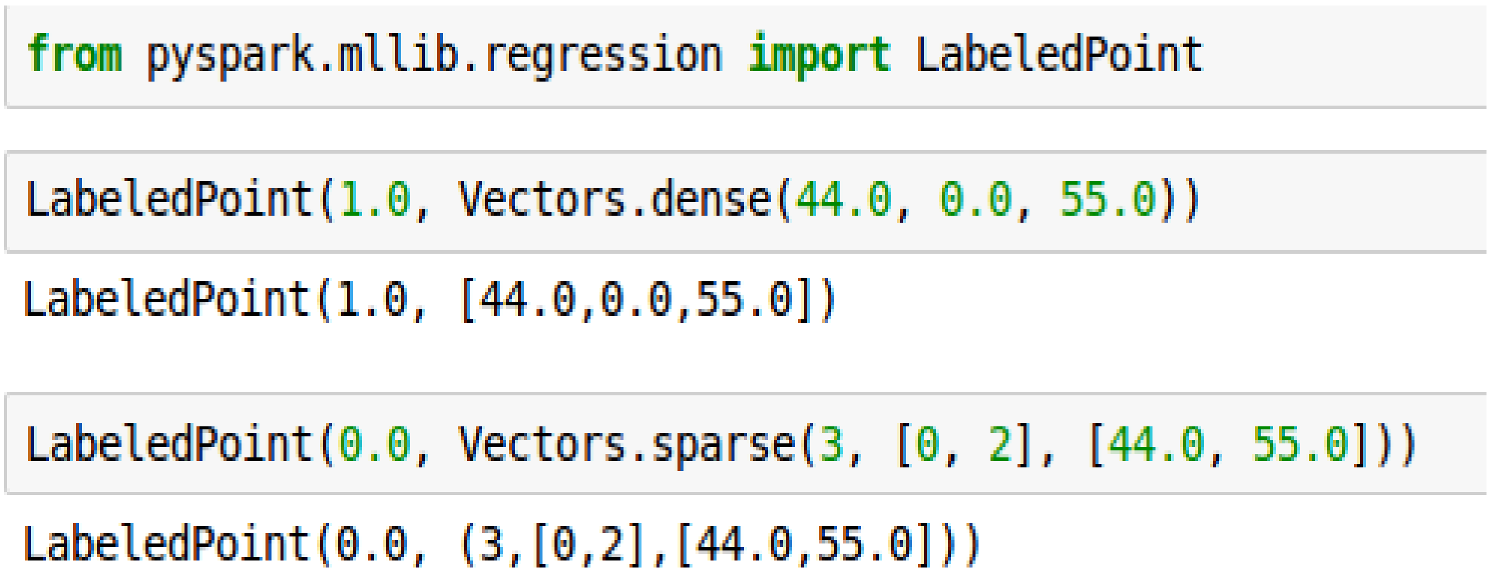
Labeled points continued

- Labels are needed for supervised machine learning
- Labels are stored as doubles so they can be used in both regression and classification problems

Labeled points continued

- In classification problems, labels must be:
  - 0 (negative) or 1 (positive) for binary classification
  - class indices starting from zero (0, 1, 2...) for multiclass

`LabeledPoint` Examples



```
from pyspark.mllib.regression import LabeledPoint
```

```
LabeledPoint(1.0, Vectors.dense(44.0, 0.0, 55.0))
```

```
LabeledPoint(1.0, [44.0,0.0,55.0])
```

```
LabeledPoint(0.0, Vectors.sparse(3, [0, 2], [44.0, 55.0]))
```

```
LabeledPoint(0.0, (3,[0,2],[44.0,55.0]))
```

Lesson Summary

- Having completed this lesson, you should be able to:
  - Understand local vectors and labeled points
  - Create dense and sparse vectors
  - Create labeled points