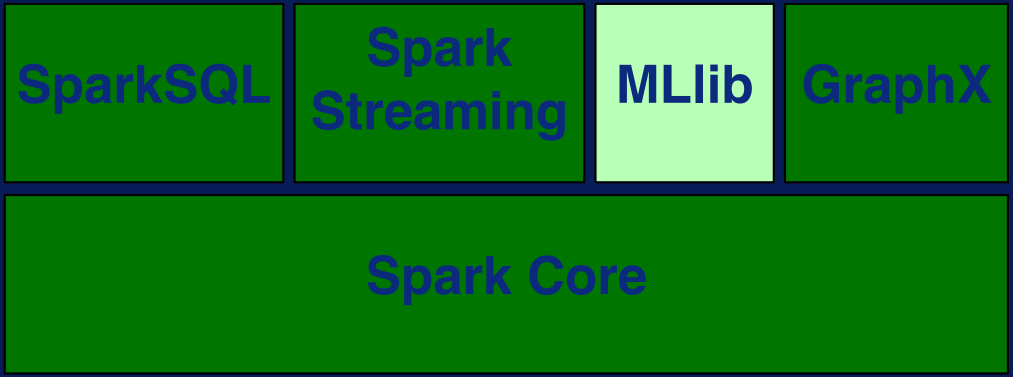
Spark MLlib

After this video you will be able to..

- Describe what MLlib is
- List main categories of techniques available in MLlib.
- Explain code segments containing MLlib algorithms.

Spark MLlib



SparkSQL

Spark Streaming

MLlib

GraphX

Spark Core

- Scalable machine learning library
- Provides distributed implementations of common machine learning algorithms and utilities
- Has APIs for Scala, Java, Python, and R

MLlib Algorithms & Techniques

- Machine Learning
  - Classification, regression, clustering, etc.
  - Evaluation metrics
- Statistics
  - Summary statistics, sampling, etc.

correlations to sample dataset

mean, stdev, etc ;

- Utilities
  - Dimensionality reduction, transformation, etc.

methods for preprocessing the data

MlLib Example – Summary Statistics

- Compute column summary statistics

```
from pyspark.mllib.stat import Statistics

# Data as RDD of Vectors
dataMatrix = sc.parallelize([ [1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12] ])

# Compute column summary statistics.
summary = Statistics.colStats(dataMatrix)
print(summary.mean())
print(summary.variance())
print(summary.numNonzeros())
```

1 → import

2 create RDD of vectors with data → each vector → a column in a data matrix

3 invokes column stats function to compute summary statistics for each column

4 → print mean, variance and number of non-zero entries for each column

MLlib Example – Classification

6 steps to:

- Build decision tree model for classification

```
from pyspark.mllib.tree import DecisionTree, DecisionTreeModel
from pyspark.mllib.util import MLUtils

# Read and parse data
data = sc.textFile("data.txt")

# Decision tree for classification
model = DecisionTree.trainClassifier
                (parsedData, numClasses=2)
print(model.toDebugString())
model.save(sc, "decisionTreeModel")
```

1 → import decision tree module

2 → import MLutils module

3

4 → build decision tree to classify in two classes

5 → print model

6 → save model in a file

MLlib Example – Clustering

- Build k-means model for clustering

```
from pyspark.mllib.clustering import KMeans, KMeansModel
from numpy import array
```

1 imports

2

```
# Read and parse data
data = sc.textFile(“data.txt”)
parsedData = data.map(lambda line:
           array([float(x) for x in line.split(‘ ‘)]))
```

using space as limiter

3

```
# k-means model for clustering
clusters = Kmeans.train (parsedData, k=3)
```

kmeans built by dividing the parsedData into 3 clusters

4

```
print(clusters.centers)
```

5 cluster centers printed out

Main Take-Aways

- MLlib is Spark’s machine learning library.
  - Distributed implementations
- Main categories of algorithms and techniques:
  - Machine learning
  - Statistics
  - Utility for ML pipeline