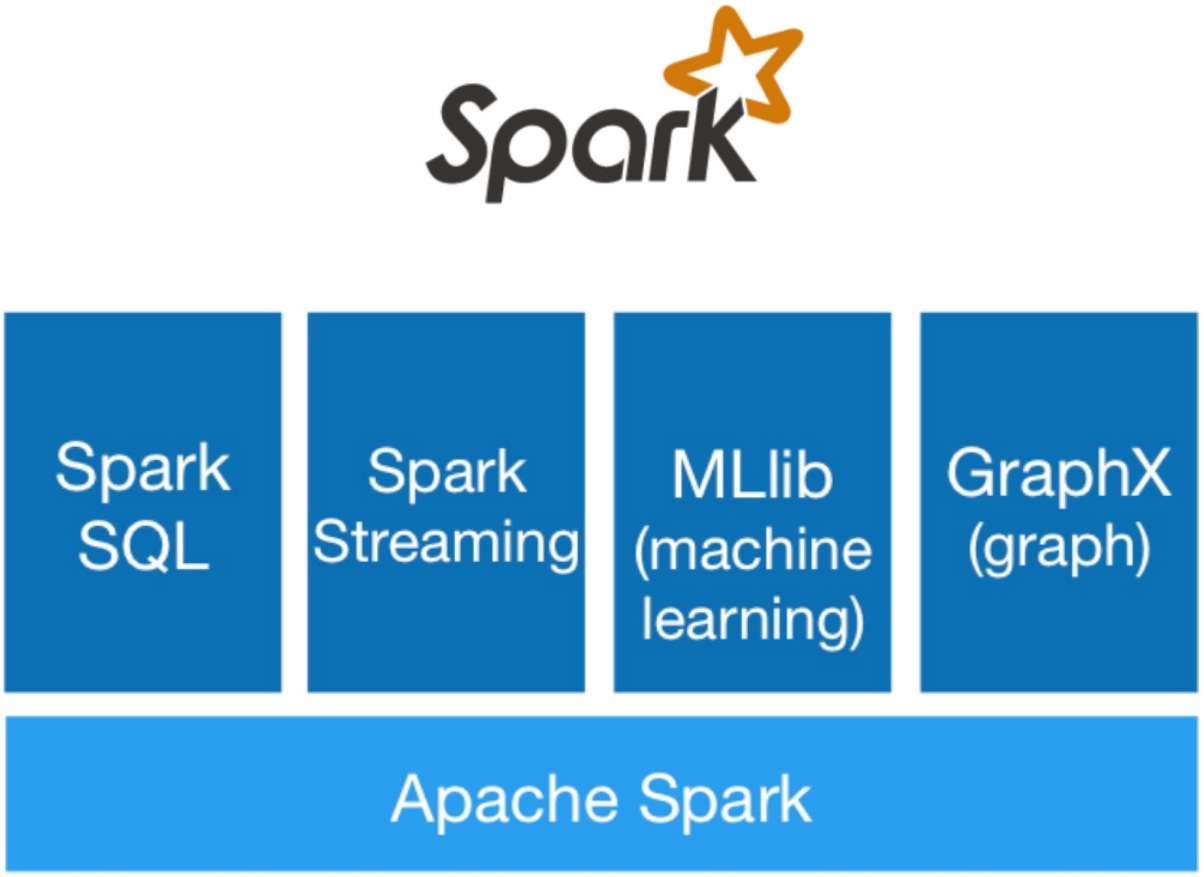
Spark MLlib Introduction

Spark MLlib Makes Machine Learning Scalable and Easy



Spark MLlib

Ease of Use

Usable in Java, Scala, Python, and R.

MLlib fits into Spark's APIs and interoperates with NumPy in Python (as of Spark 0.9) and R libraries (as of Spark 1.5). You can use any Hadoop data source (e.g. HDFS, HBase, or local files), making it easy to plug into Hadoop workflows.

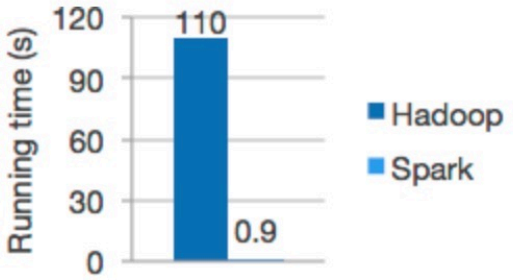
```
data = spark.read.format("libsvm")\
  .load("hdfs://...")

model = KMeans(k=10).fit(data)
```

Calling MLlib in Python

Performance

High-quality algorithms, 100x faster than MapReduce.

Spark excels at iterative computation, enabling MLlib to run fast. At the same time, we care about algorithmic performance: MLlib contains high-quality algorithms that leverage iteration, and can yield better results than the one-pass approximations sometimes used on MapReduce.



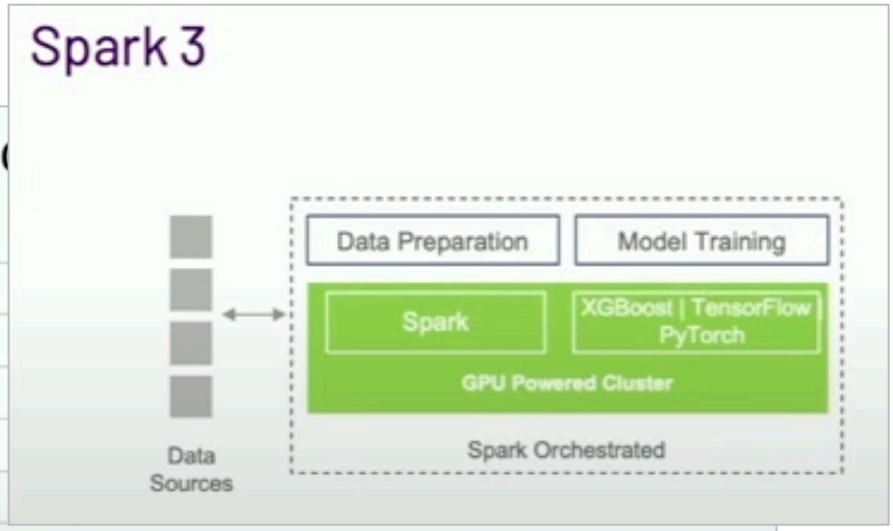
Logistic regression in Hadoop and Spark

Runs Everywhere

Spark runs on Hadoop, Apache Mesos, Kubernetes, standalone, or in the cloud, against diverse data sources.

Spark MLlib

- GPU is going to be supoprted in Spark 3.x for both ETL (e.g. in Spark SQL) and Machine Learning (e.g. in Spark MLlib)



	Original CPU (1 Core for ETL, 96 Core CPU for Training)	Spark CPU (96 Core for ETL & Training)	Spark CPU (96 Core for ETL) & Spark GPU (1-V100 Training)	Spark GPU (8-V100 for ETL & 1-V100 Training)
Training	45.0	45.0	0.7	0.7
ETL	144.0	12.1	12.1	0.5

Why is **Spark MLlib**?

Two major use cases for using Spark MLlib:

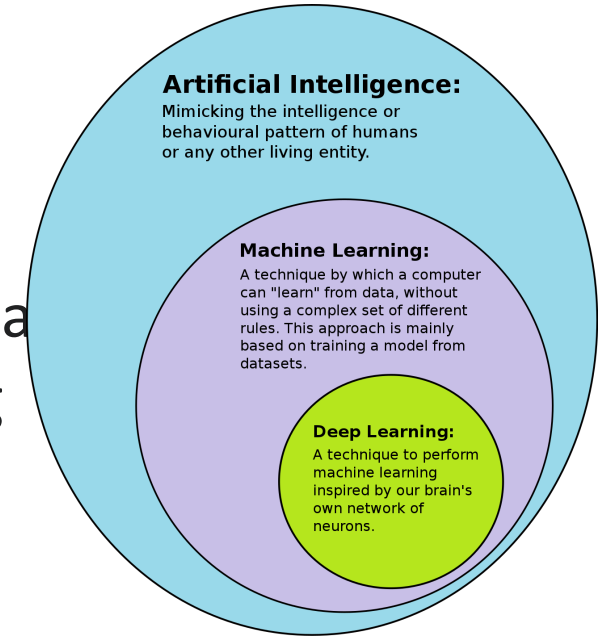
1. **When the input data become too difficult to put on a single machine, use Spark to do the heavy lifting. Spark makes distributed machine learning very simple, scalable and fast**.

2. Use Spark/Spark MLlib for preprocessing and feature generation from a large amount of data to reduce the amount of time it might take to produce training and test sets. Then use single-machine learning libraries to train on those data sets.

(compared to other ML libraries like *s**cikit-lear**n,* ***tensorflow***, etc.)

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of AI.

Machine learning algorithms build a mathematica model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so.

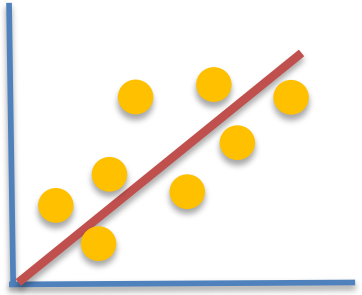


https://en.wikipedia.org/wiki/Machine_learning

Types of ML Problems

Supervised Learning

- Classification
  - Giving a data point, predict its label (or class)
    - Gender prediction: *male or female*
    - Spam detection: *spam or not spam(ham*)
    - Purchase preidction: *purchase or not*
    - Sentiment analysis: *positive or negative*
- Regression
  - Giving a data point, prdict a continious value
    - Demand forcast *for 2020/12*
    - Revenue forcast *for 2021*
- Recommandation
  - *Find out what products to recommand for a user?*
  - *Find out what users to recommand for a product?*

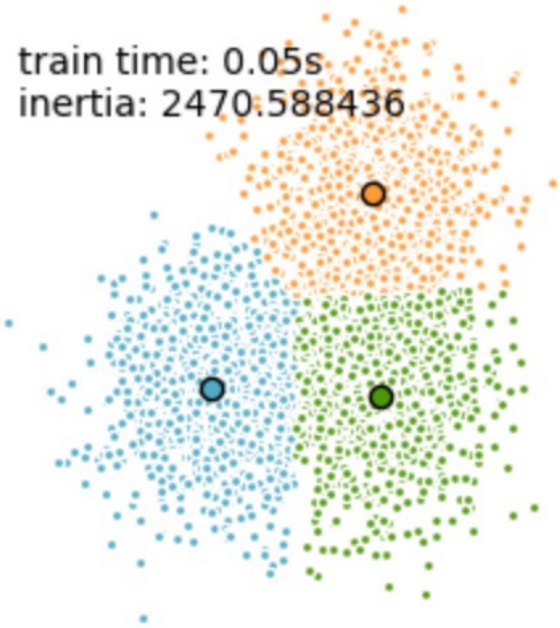


	Prod 1	Prod 2	Prod 3	Prod N
User1	5	3	4	3
User2	3	4	4	5
User3	5	3	4	4
UserN	3	3	4	5

Types of ML Problems

Unsupervised Learning

- Clustering
  - Let the algorithem assign the groups for data points
    - Fraud detection *for transacations*
    - Theme detection *for articles*
    - *Outliers detection*



The Examples of Machine Learning

Supervised Learning

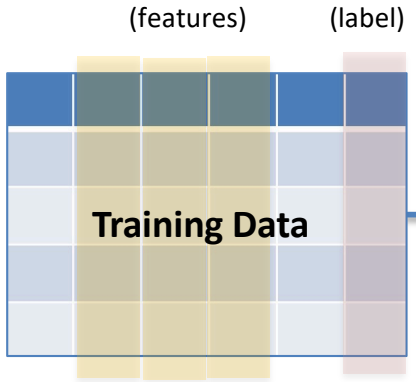
[Problem to solve] **Detect a message as spam or not spam** **(A classification problem)**

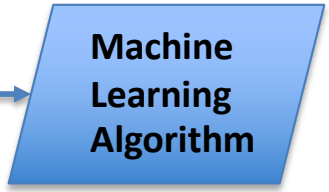
1. Training phase

Messages data

Features:
- *Message*
- *Content length*
- *Sender IP*
- *….*

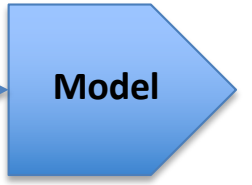
Label:
- *Spam / Ham*





Classification algorithms

e.g. Decision tree, Random forest, SVM, Logistic Regression, ANN, etc.



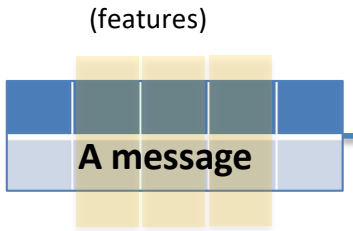
- A set of rules (if..else..)
- Equations with learnec coefficients
- Vectors, matrixes

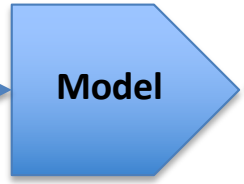
A Classification model

e.g. a decision tree

2. Prediction phase

An email to detect





A Spam

The Examples of Machine Learning

Supervised Learning

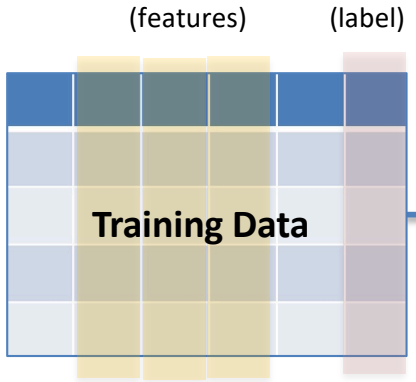
[Problem to solve] **Demand forecast for certain condition** **(A regression problem)**

1. Training phase

Sales data

Features:
- *Product category*
- *User location*
- *Date of week*
- *Holiday or not*
- *Promotion or not*

Label:
- *Sales*



Machine Learning Algorithm

Regression algorithms

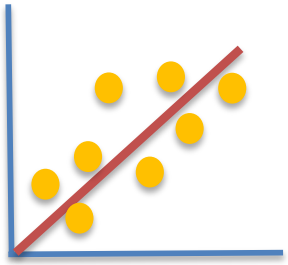
e.g. Linear Regression, Tree-based Regressors, SVM, ANN, etc.

Model

- A set of rules (if. Else..)
- Equations with learned coefficients
- Vectors, matrixes

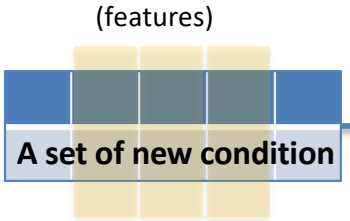
A Regression model

e.g. a linear equation with learned coefficients

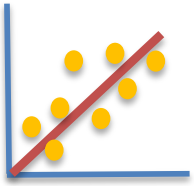


2. Prediction phase

Make a demand forecast



Model



10K items

The Examples of Machine Learning

Supervised Learning

[Problem to solve] **Recommend movies/products to users** (**A recommendation problem**)

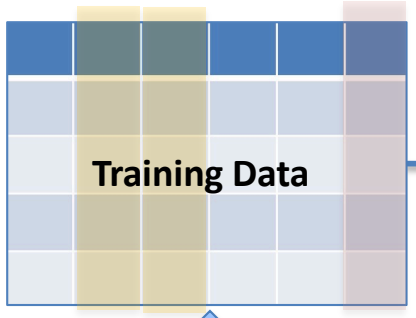
1. Training phase

Sales data

Features:
- *User*
- *Movie*

Label:
- *Rating*

(features) (label)



	Prod 1	Prod 2	Prod 3	Prod N
User1	5	3		3
User2	3		4	
User3		3	4	
UserN	3			5

Machine Learning Algorithm

Recommendation algorithms

e.g. ALS, KNN, etc.

Model

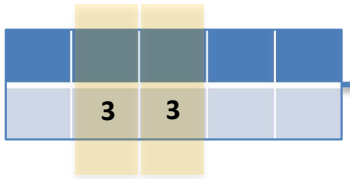
- Vectors, matrixes

A Recommendation model

	Prod 1	Prod 2	Prod 3	Prod N
User1	5	3	4	3
User2	3	4	4	5
User3	5	3	4	4
UserN	3	3	4	5

2. Prediction phase

The user 3 watched movie watch movie 3



A set of new condition

Model

	Prod 1	Prod 2	Prod 3	Prod N
User1	5	3	4	3
User2	3	4	4	5
User3	5	3	4	4
UserN	3	3	4	5

Top Recommended: movie 1

The Examples of Machine Learning

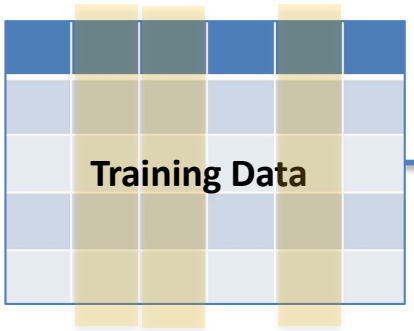
Unsupervised Learning

[Problem to solve] **Automatically group similar news contents for further identifying their themes**
(**A clustering problem**)

News Article

Features:
- *Title*
- *Article*
- *Author*

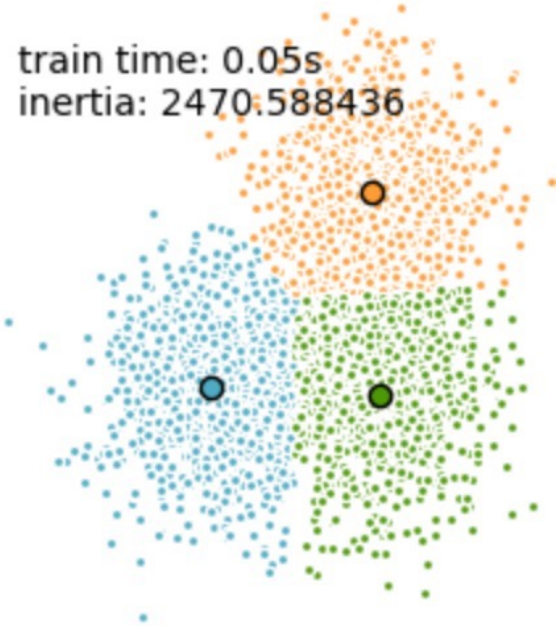
(features)



Machine Learning Algorithm

Clustering algorithms

e.g. Kmeans++, etc.



Next step:
To evaluate each group by looking at the contents in the same group and different groups to have a clue how to label articles

Spark MLlib API Covers

- Data Preprocessing
- Feature Engineering
  - Feature transforming, extraction,…
- Distributed Machine Learning Alogrithms
  - Regression: *Linear Regression, Generalized Linear Regression*,…
  - Classfication: *Logistic Regression, Decision trees, Random Forest, …*
  - Clustering: *K-means*, …
  - Recommendation : ALS (Netflix)
  - Association Rules Mining: Parallel FP-growth,…
  - …
- Performance Evaluation

Spark MLlib Versions

pyspark.ml (current)
pyspark.mllib (maintaince)

- pyspark.ml package
  - ML Pipeline APIs
  - pyspark.ml.param module
  - pyspark.ml.feature module
  - pyspark.ml.classification module
  - pyspark.ml.clustering module
  - pyspark.ml.functions module
  - pyspark.ml.linalg module
  - pyspark.ml.recommendation module
  - pyspark.ml.regression module
  - pyspark.ml.stat module
  - pyspark.ml.tuning module
  - pyspark.ml.evaluation module
  - pyspark.ml.fpm module
  - pyspark.ml.image module
  - pyspark.ml.util module
- pyspark.mllib package
  - pyspark.mllib.classification module
  - pyspark.mllib.clustering module
  - pyspark.mllib.evaluation module
  - pyspark.mllib.feature module
  - pyspark.mllib.fpm module
  - pyspark.mllib.linalg module
  - pyspark.mllib.linalg.distributed module
  - pyspark.mllib.random module
  - pyspark.mllib.recommendation module
  - pyspark.mllib.regression module
  - pyspark.mllib.stat module
  - pyspark.mllib.tree module
  - pyspark.mllib.util module

pyspark.mllib	pyspark.ml
Older	Newer
Directly based on RDD	Mainly based on DataFrame
Not support ETL, chaining transformations	ETL, chaining transformations
No pipeline support	Support for ML pipelines
Good performance	Evern better performance
compatibility with 1.x applications	Not compatialbe with 1.x applications

ML Training Pipeline in Spark MLlib (Supervised Learning)

Spark SQL API for Python: **pyspark.sql + pyspark.ml**

1 **Create DataFrame** 2 **Preprocessing and Feature engineering** 3 **Split train and test set** 4 **Model training**

1 **Structural/Semi-Structural Files**

HDFS/S3/…

1. CSV
2. JSON files
3. Parquet files
4. Avro files

…

```
people = spark.read.json(…)
```

2 **RDB Table**

```
people = spark.read.jdbc(…)
```

RDB

3 **RDD + Schema**

[(1, “John”, 20), (2, “May”, 17), …]

(type: RDD)

4 **Driver-local list + Schema**

[(1, “John”, 20), (2, “May”, 17), …]

(type: list)

1 **Make queries in SQL**

2 **Use DataFrame Transformation API**

3 **Feature Engineering Classes for feature transformation, extraction, etc.**

e.g.

- Categorical value to numerical value
- Standardize/normalize value
- Tokenize, stop words removal, TF-IDF …

data
(DataFrame)

label features

final_data
(DataFrame)

Use an ML Algorithm Classes to train data and return the model

label features

train_data
(DataFrame)

model
(Transformer)

model
(Transformer)

transform

label features prediction

predicted_result
(DataFrame)

Performance evaluation

label features

test_data
(DataFrame)

6 **Continuous tuning and Model selection**

5 **Test data prediction and Evaluation**

7 **Save model for future use**

ML Prediction Pipeline in Spark MLlib (Supervised Learning)

Spark SQL API for Python: **pyspark.sql + pyspark.ml**

1 **Create DataFrame** 2 **Preprocessing and Feature engineering** 3 **Make prediction** 4 **Output result**

1 **Structural/Semi-Structural Files**

HDFS/S3/…

1. CSV
2. JSON files
3. Parquet files
4. Avro files

…

```
people = spark.read.json(…)
```

2 **RDB Table**

```
people = spark.read.jdbc(…)
```

RDB

3 **RDD + Schema**

[(1, “John”, 20), (2, “May”, 17), …]

(type: RDD)

4 **Driver-local list + Schema**

[(1, “John”, 20), (2, “May”, 17), …]

(type: list)

data
(DataFrame)

1 **Make queries in SQL**

2 **Use DataFrame Transformation API**

3 **Feature Engineering Classes for feature transformation, extraction, etc.**

e.g.

- Categorical value to numerical value
- Standardize/normalize value
- Tokenize, stop words removal, TF-IDF …

features

data_to_predict
(DataFrame)

5 transform

model
(Transformer)

features prediction

predicted_result
(DataFrame)

6 output

0 Load the previously saved model

0 **Load the saved model anytime before use**

ML Pipeline in Spark MLlib (Unsupervised Learning)

Spark SQL API for Python: **pyspark.sql + pyspark.ml**

1 **Create DataFrame** 2 **Preprocessing and Feature engineering** 3 **Model training** 4 **Make prediction**

1 **Structural/Semi-Structural Files**

HDFS/S3/…

1. CSV
2. JSON files
3. Parquet files
4. Avro files

…

```
people = spark.read.json(…)
```

2 **RDB Table**

```
people = spark.read.jdbc(…)
```

RDB

3 **RDD + Schema**

[(1, “John”, 20), (2, “May”, 17), …]

(type: RDD)

4 **Driver-local list + Schema**

[(1, “John”, 20), (2, “May”, 17), …]

(type: list)

data
(DataFrame)

1 **Make queries in SQL**

2 **Use DataFrame Transformation API**

3 **Feature Engineering Classes for feature transformation, extraction, etc.**

e.g.

- Categorical value to numerical value
- Standardize/normalize value
- Tokenize, stop words removal, TF-IDF …

features

final_data
(DataFrame)

Use an ML Algorithm Classes to train data and return the model

transform

model
(Transformer)

features prediction

predicted_result
(DataFrame)

7 output

(performance evaluation)

6 **Continuous tuning and Model selection** 5 **Evaluation**

7 **Save model for future use**

Spark Program Run-time Excution Flow (In Spark MLlib API)

Spark SQL API for Python:
pyspark.sql

Tasks are internal instructions according to your codes

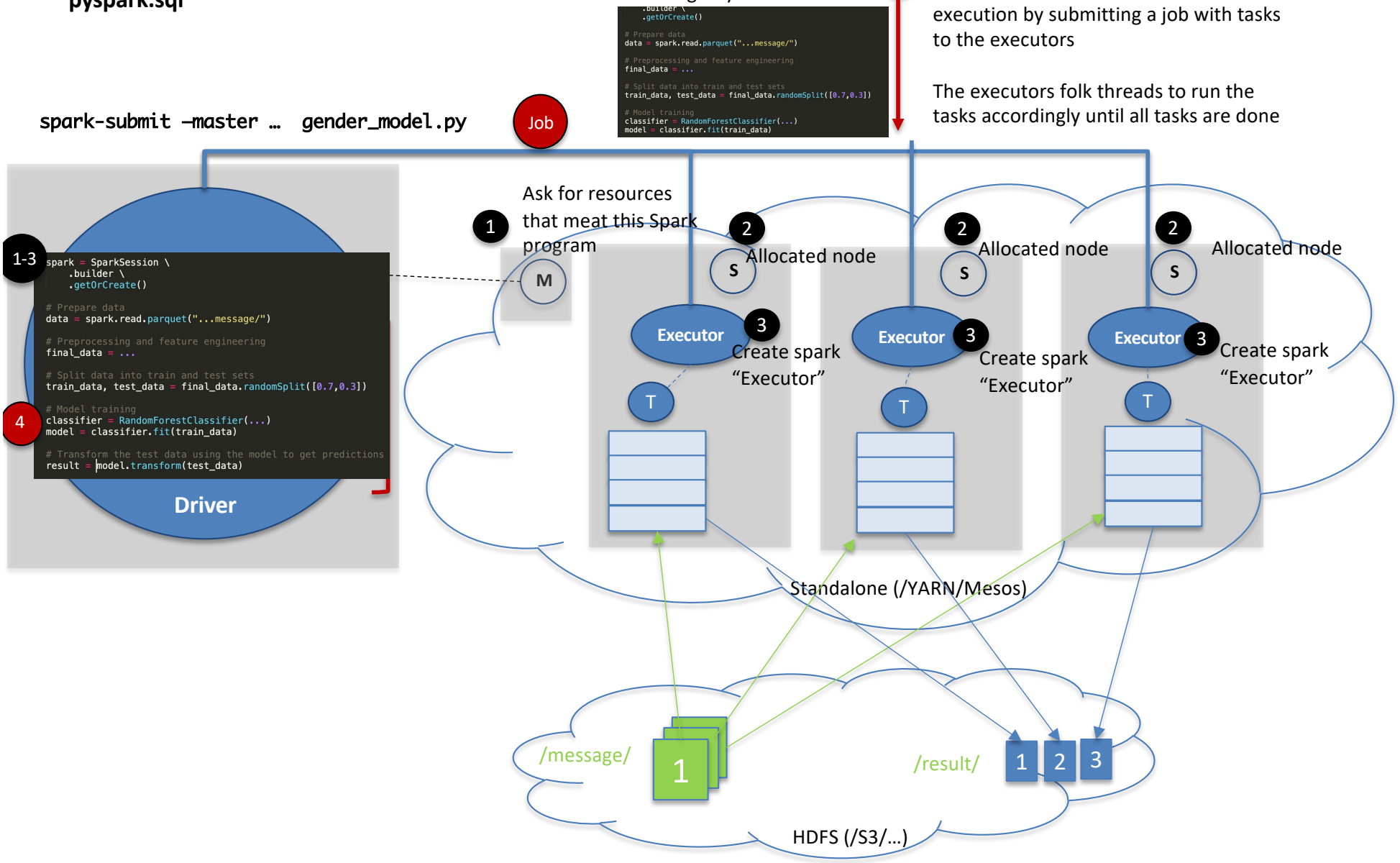
5

DataFrame's action method trigger real execution by submitting a job with tasks to the executors

The executors folk threads to run the tasks accordingly until all tasks are done

```
spark-submit –master …  gender_model.py
```

Job



Transformers and Estimators

Feature Engineering (Estimators)

- StringIndexer
- OneHotEncoder
- StandardScaler
- CountVectorizer
- IDF
- …

Feature Engineering (Transformers)

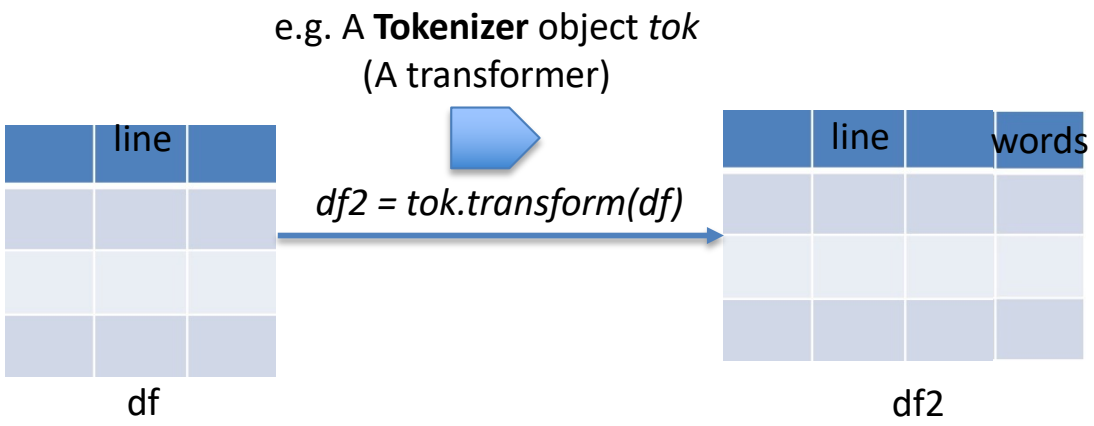
- Tokenizer
- RegexTokennizer
- Normalizer
- VectorAssembler

ML Algorithms (Estimators)

- DecisionTreeClassifier
- RandomForeastClassifer
- LogicticRegression
- LinearSVC
- LinearRegression
- DecisionTreeRegresspor
- RandomForestRegressor
- ALS
- Kmean++
- …

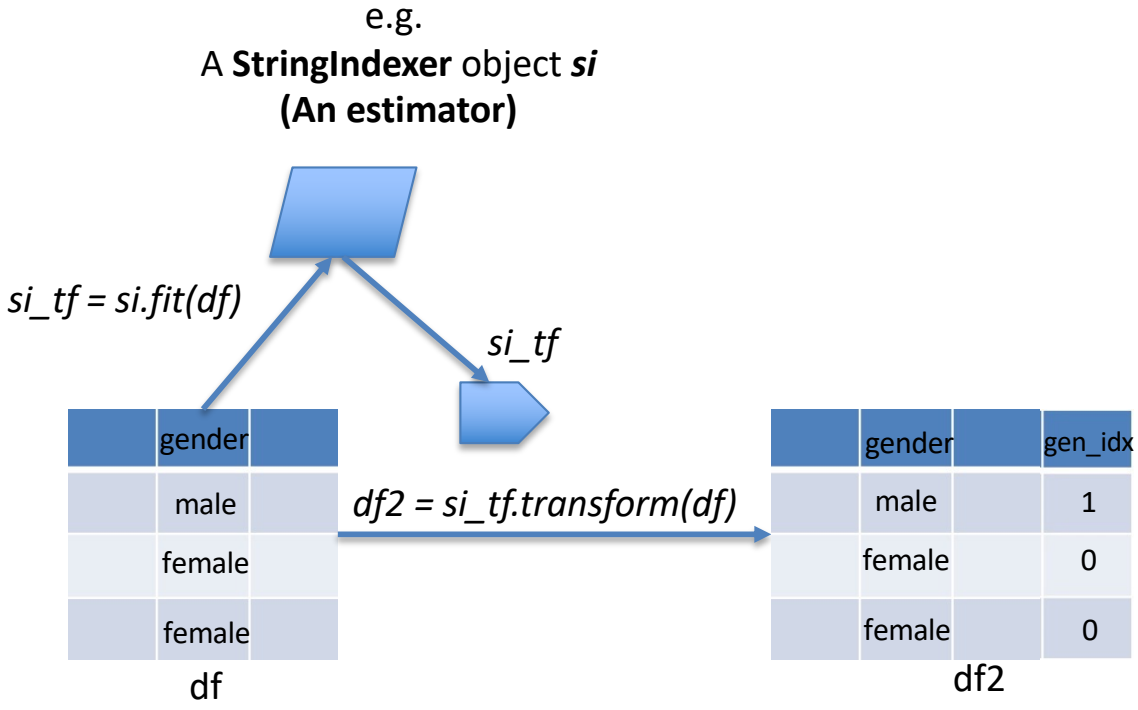
Transformers

- **The Transformer** has a **.transform()** method and return a transformed DataFrame



Estimators

- **The Estimator** has a **.fit(df)** method to train data and then **return a transformer** for furhtur transoforming the **df**



Dive-in Demos: Spark MLlib Usage

logs_201905.csv

```
device_id,timestamp,product_category_id,ip,lat,lon,device_type,connect_type,age_group,gender,member_id
2eecf6cc-b2d8-f13b-4d18-93835c92f2a9,2019-05-01T08:00:04.000+08:00,132,114.137.185.183,25.027023,121.41819,1,2,"","",c55941d974fa7f9cdaad51e52cc2cb3e
7f9b4249-b0df-39de-4424-326d680f30e8,2019-05-01T08:00:07.000+08:00,38,115.82.34.79,25.02705,121.557396,2,2,"","",d578b2f0aa6e00f63865825e99ab70ea
db396d28-b748-56ab-4643-8f49a8caded8,2019-05-01T08:00:12.000+08:00,17,49.216.141.189,24.090103,120.73134,2,2,"","",5f27b787212e1690011599221 95a9e23
0cabdfb3-b98e-9fbd-49ae-c232f8cd5b8b,2019-05-01T08:00:15.000+08:00,60,27.52.70.89,24.918932,121.19233,2,1,36-50,0,adb6b4fc4af10ff6840ebcbe16fe782d
1e232f10-8fa5-d175-4e41-39963b7bd827,2019-05-01T08:00:19.000+08:00,156,223.136.185.112,25.06417,121.426994,1,2,over 50,1,2af73f9c2f39b2ad0825cb23102301d5
0a95869d-aa7c-0799-4c83-46154e85adee,2019-05-01T08:00:19.000+08:00,104,27.246.4.74,24.696573,121.80277,1,1,"","",ca6486a6f6d2b9b49e4397e99179019e
9804ffa2-b4b3-73cb-4d99-a452ee27d076,2019-05-01T08:00:20.000+08:00,53,2001:b400:e269:eb93:79db:2898:5df0:ed92,25.129053,121.7445,2,2,36-50,0,88822a61c8cf27d9580b3d0af3c9bfab
b2d81105-baac-aed9-490b-732f0eafcbbd,2019-05-01T08:00:20.000+08:00,6,27.246.170.248,22.95496,120.301216,1,2,"","",97e6cb863f23b913ff6739acbccb1620
8048c12c-b7cc-84d4-4252-3e65e9d2898a,2019-05-01T08:00:23.000+08:00,106,101.137.154.116,24.93326,121.29555,1,2,"","",76fc22be6cc209088f2838ccd9fdb9c3
0b146116-8876-33b1-4825-7ad66fce5c18,2019-05-01T08:00:23.000+08:00,11,223.136.239.129,25.008995,121.479866,2,2,"","",b0ddf7e8c19f6f175a29f0b7096daac2
5278c4ca-be80-21fe-4cd1-2fb677e0e721,2019-05-01T08:00:24.000+08:00,113,59.115.112.164,25.11713,121.527435,1,2,21-35,1,177e57bfd282151952cbc5327f3a1d52
985f29ee-ba58-6b2f-4a51-93d0f9f5bace,2019-05-01T08:00:27.000+08:00,104,39.10.197.228,24.98072,121.54427,2,2,21-35,1,5e94847dacd305725ed7fb215fa7f7a7
b01fa9ff-b804-d917-4465-c6a68ce4581d,2019-05-01T08:00:38.000+08:00,3,2402:7500:539:379:39ed:9596:ba9d:f689,24.9945,121.5235,2,1,"","",54b626b3b6a74193195935237f49f7b9
bea679fd-af94-5d9b-4f15-b28e97c3fc08,2019-05-01T08:00:41.000+08:00,6,101.136.237.118,25.066343,121.530396,2,2,under 20,0,fad7773c3ce00c2297cc652c7bb94abe
de7aac5a-babf-2534-43e7-ffecdebfb77d,2019-05-01T08:00:45.000+08:00,39,110.28.65.94,22.61878,120.28969,2,2,"","",2a907c67a5007861347b0a61e024d32e
e1cebe0d-a94e-a953-4aee-befa646f073b,2019-05-01T08:00:48.000+08:00,41,2401:e180:8821:be9f:7b8e:8d7c:1b6a:3dd,24.0721,120.43806,2,1,"","",4e6e4c58df5bbd12bf9adb18a5373363
d3f78cc5-a8fb-1170-45ce-679ea679ad54,2019-05-01T08:00:49.000+08:00,12,39.9.39.103,24.1759,120.64938,2,2,"","",2e86f48e8a67732c990a693d4480c068
2ad6a032-93c4-082f-4bea-c83f85821d27,2019-05-01T08:00:50.000+08:00,147,2001:b400:e27e:54bd:d757:35be:2c24:c7d6,24.994648,121.51234,1,1,"","",7b7319e6a63ba7bbf5fddd10d2982534
417d6ce7-84bb-5bfa-40cb-a080367264d1,2019-05-01T08:00:52.000+08:00,123,2001:b400:e78f:7348:1475:d520:7b02:395d,22.650072,120.33544,2,1,21-35,1,a608624685fb00a6fcf193da47626dc2
22866a34-ac33-9337-4312-1e5546c84e6d,2019-05-01T08:00:53.000+08:00,27,101.136.73.38,22.558563,120.5574,1,1,under 20,0,fc4b5e5ba21d9d7522c79009073d858a
a53449cf-b2d3-d83b-4d37-1cf4c16513cd,2019-05-01T08:00:56.000+08:00,16,49.219.64.184,23.919825,120.539604,1,2,"","",12ef1f7a6c39cfb6b95e104a06fb73c8
43c14e9b-8b94-0fd2-47eb-41156ff79d8a,2019-05-01T08:00:57.000+08:00,101,2001:b400:e75a:9ff2:cf5a:19e6:2ebb:bc27,22.581116,120.31894,1,1,"","",0173c4900868a8c20d631e5a2dc049ff
```

```
Database changed
mysql> show tables;
+----------------------------
| Tables_in_ec_web_logs_analy
+----------------------------
| device_inferred_age_group
| device_inferred_gender
| device_inferred_location
| stats_device_hourly_activit
| stats_inferred_age_group_co
| stats_inferred_gender_count
+----------------------------
6 rows in set (0.00 sec)
```

MySQL

../raw

../data

ec_web_logs

Gender prediction model

edict (label) all
gender info

Gender distribu

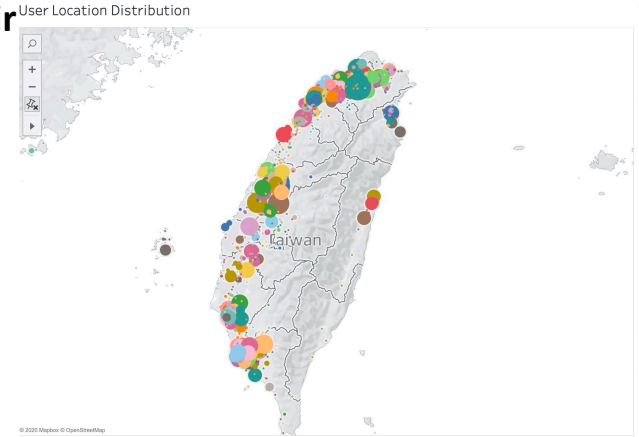
Age Group
Prediction model

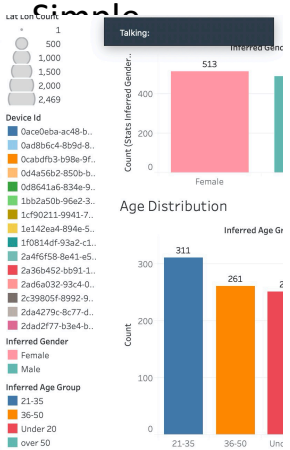
Predict (label) all users'
age groups

Age group distribution

Clustering users' location
points to summrize their
major points of visits

User activities





More Demos on Spark MLlib Usage

- Movie recommandation
- Clustering on titanic dataset
- Feature engineering for NLP
- Spam detection model

Reference

Classification Model Evaluation

- Confusion Matrix

		Predicted label	
		Cat	Dog
Actual class	Cat	5	3
	Dog	2	3

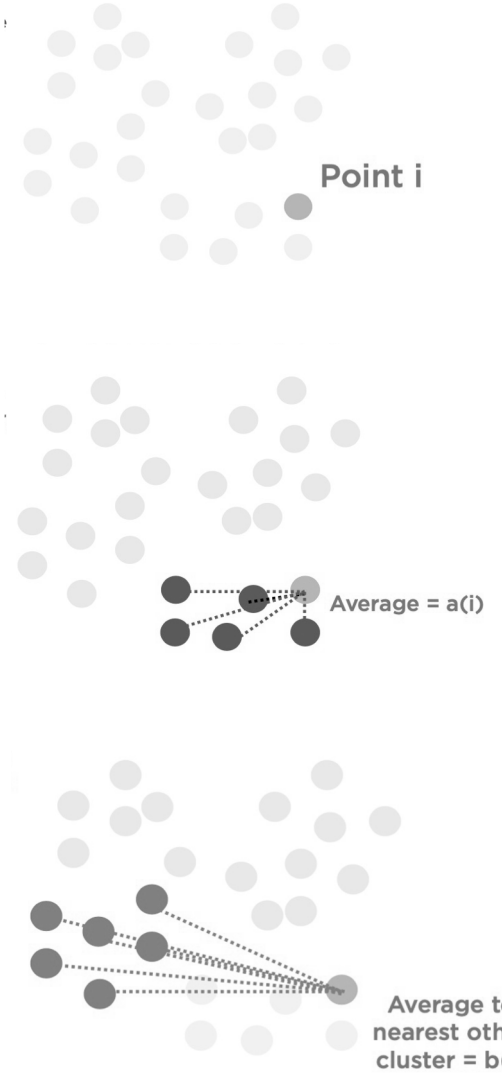
		Predicted label	
		P	N
Actual class	P	TP	FN
	N	FP	TN

P = positive
N = Negative
TP = True Positive
FP = False Positive
TN = True Negative
FN = False Negative

- Accuracy: (TP + TN) / Total Num
  - 61.53%
- Presision: TP / (TP + FP)
  - 74.42%
- Recall: TP / (TP + FN)
  - 62.5%

$$F_1 = 2 \times \frac{\text{Precision x Recall}}{\text{Precision + Recall}}$$

Clustering Model Evaluation



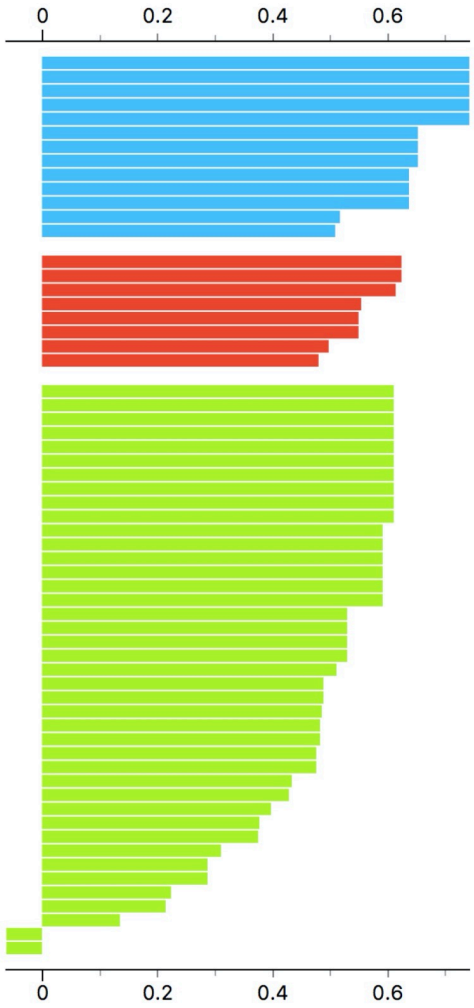
Silhouette Coefficient

For any point i

$$s(i) = \frac{b(i) - a(i)}{\text{Larger of } b(i) \text{ and } a(i)}$$

Ideally, a(i) = 0, b(i) = Infinity

Worst case, a(i) = Infinity, b(i) = 0



Recommendation in MLlib

- Alternating Least Squares (ALS)
  - https://dl.acm.org/doi/10.1109/MC.2009.263