Работа с большими данными

Содержание курса

01 Введение. Распределенные вычисления. MapReduce.

02 HDFS. Apache Spark. RDD.

03 Базы данных. Spark SQL. Хранение больших данных.

04 Подробнее о модели вычислений Spark. Знакомство со Scala.

05 Алгоритмы машинного обучения на больших данных. spark.ml.

06 Рекомендательные системы. Виды. Их метрики.

07 Обработка потоковых данных. Structured streaming и интеграция с spark.ml.

08 Модели в продакшен. Управление кластером.

5. Алгоритмы машинного обучения на больших данных. Spark.ML

План:

1. Линейная регрессия оптимизация
2. Градиентный спуск что в нем параллелить?
3. Сравнение Spark.ML с другими вариациями
4. Случайный лес, как параллелить?
5. Проблемы невоспроизводимости
6. Что позволяет Spark.ML

Линейная регрессия, её оптимизация

Вспомним! Как работает линейная регрессия

$$D = ((x, y) \mid x \in \mathbb{R}^n,\ y \in \mathbb{R})$$

$$f : x \mapsto y$$

$$\overset{*}{f} = \arg \min_{f = w \cdot x + w_0} \mathcal{L}(f, D)$$

$$\mathcal{L} = \sum_{x,y \in D} (f(x) - y)^2 \frac{1}{|D|}$$

$$\mathcal{L} = \sum_{x,y \in D} (f(x) - y)^2 \frac{1}{|D|}$$

$$\mathcal{L} = \sum_{x,y \in D} (f(x) - y)^2 \frac{1}{|D|}$$

$$= \sum_{x,y \in D} (w \cdot x + w_0 - y)^2 \frac{1}{|D|}$$

$$
\begin{aligned}
\mathcal{L} &= \sum_{x,y \in D} (f(x) - y)^2 \frac{1}{|D|} \\
&= \sum_{x,y \in D} (w \cdot x + w_0 - y)^2 \frac{1}{|D|} \\
&= \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y)^2 \frac{1}{|D|}
\end{aligned}
$$

$$(w_0, w_1, \ldots, w_n):$$
$$\sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y)^2 \to \min$$

Алгоритм градиентного спуска

```
def gradient_descent(X, y):

    epsilon = 1.0
    alpha = 1e-2
    w = random(n)

    while epsilon > 1e-4:
        grad = gradient(w, X, y)
        w = w - alpha * grad
        epsilon = np.linalg.norm(grad)

    return w
```

Как будем считать градиент?

$$\nabla \mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial w_0}, \frac{\partial \mathcal{L}}{\partial w_1}, \ldots, \frac{\partial \mathcal{L}}{\partial w_n} \right)$$

$$\mathcal{L} = \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y)^2 \frac{1}{|D|}$$

$$\frac{\partial \mathcal{L}}{\partial w_i} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y) \cdot x_i$$

$$\frac{\partial \mathcal{L}}{\partial w_0} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y) \cdot x_1$$

$$\cdots$$

$$\frac{\partial \mathcal{L}}{\partial w_n} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y) \cdot x_n$$

$$\frac{\partial \mathcal{L}}{\partial w_0} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y) \cdot x_1$$

$$\cdots$$

$$\frac{\partial \mathcal{L}}{\partial w_n} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y) \cdot x_n$$

$$O\left(n^2 |D|\right)$$

$$\frac{\partial \mathcal{L}}{\partial w_0} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y)$$

$$\frac{\partial \mathcal{L}}{\partial w_1} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y) \cdot x_1$$

$$\cdots$$

$$\frac{\partial \mathcal{L}}{\partial w_n} \propto \sum_{x,y \in D} (x_1 w_1 + x_2 w_2 + \cdots + x_n w_n + w_0 - y) \cdot x_n$$

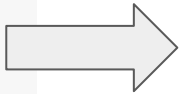
$$O\left(n|D|\right)$$

```python
def gradient_descent(X, y):

    epsilon = 1.0
    alpha = 1e-2
    w = random(n)

    while epsilon > 1e-4:
        grad = gradient(w, X, y)
        w = w - alpha * grad
        epsilon = np.linalg.norm(grad)

    return w
```



```python
def gradient(X, y, w):

    def mapper(x, y):
        return x * (x.dot(w) - y)

    return run_map_reduce(
        X, y
        mapper=mapper,
        reducer=sum,
    )
```

Spark way

```
def gradient_descent(x_y):

    epsilon = 1.0
    alpha = 1e-2
    w = random(n)
    x_y.cache() # Всего-то делов

    while epsilon > 1e-4:
        grad = x_y.map(mapper).sum() # Всего-то делов
        w = w - alpha * grad
        epsilon = np.linalg.norm(grad)

    return w
```

Градиентный спуск вариации

https://habr.com/ru/company/ods/blog/327250/

Batch gradient descent

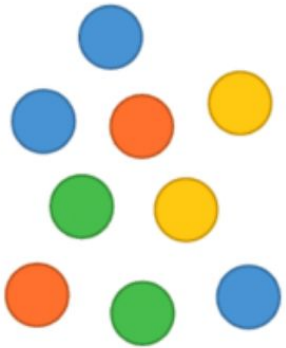
Mini-batch gradient Descent

Stochastic gradient descent

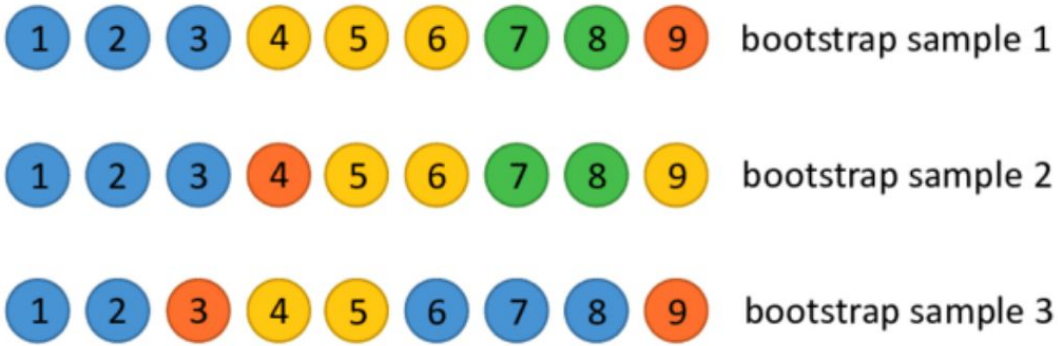
Случайный лес

https://habr.com/ru/company/ods/blog/324402/

Bootstrap sample



original sample



Невоспроизводимость сохранения в Spark

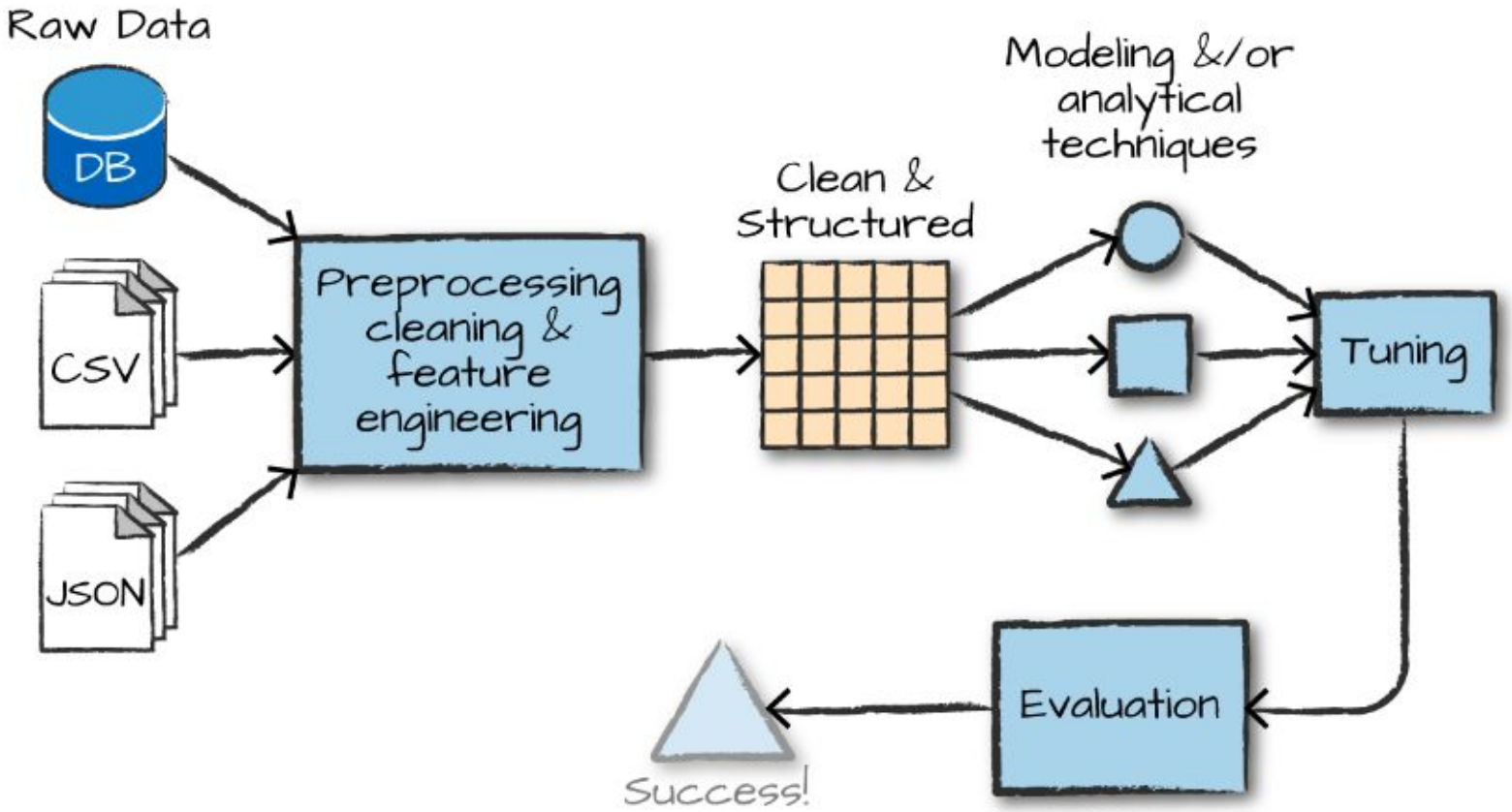
```
[1 2 3] [4 5 6 7] [8 9 10]
```

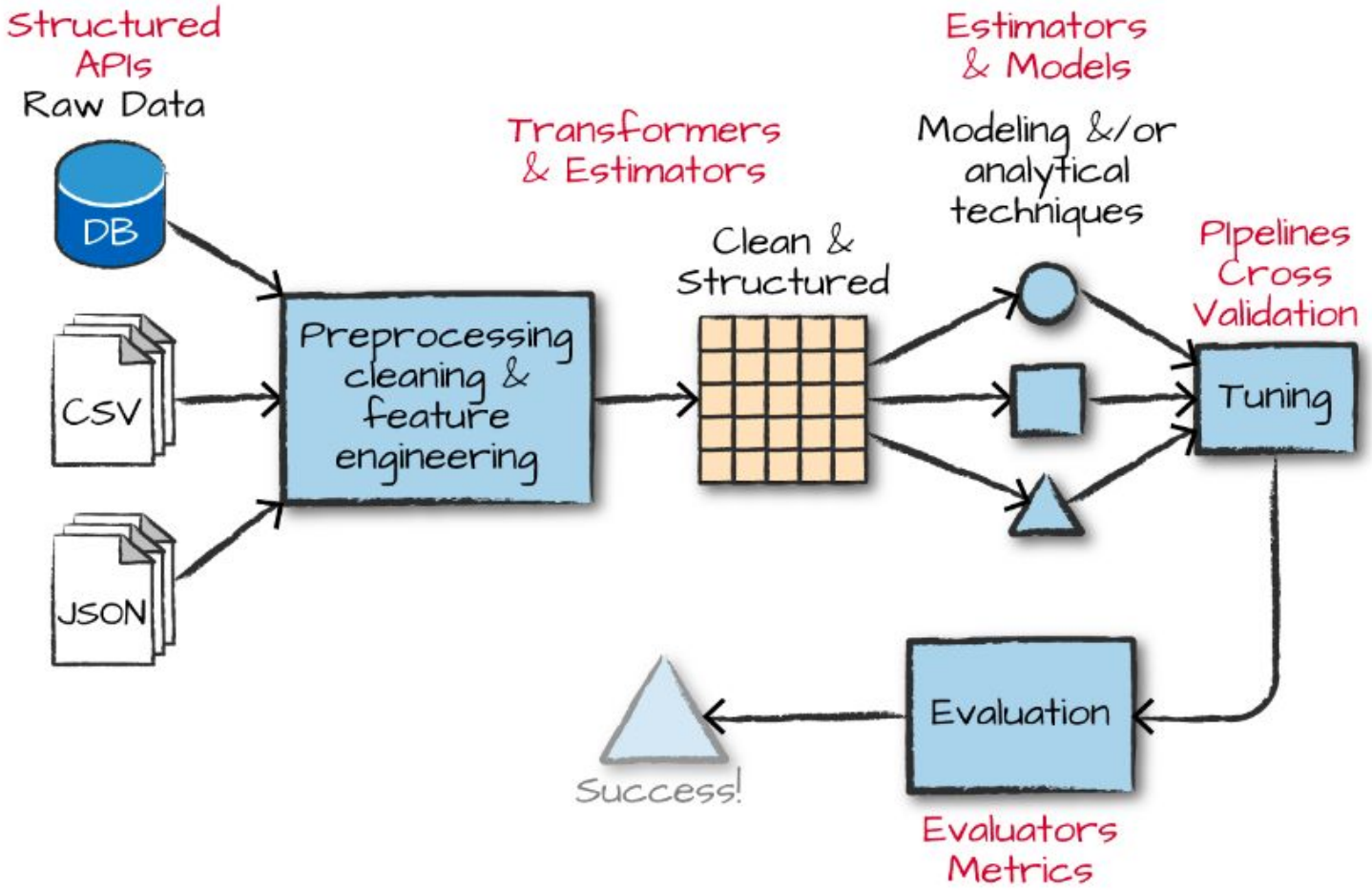
Пересохраним данные

```
[1 2] [3 4 5] [6 7 8 9 10]
```

Что позволяет Spark.ML

http://spark.apache.org/docs/latest/ml-features.html


Raw Data
DB
CSV
JSON
Preprocessing cleaning & feature engineering
Clean & Structured
Modeling &/or analytical techniques
Tuning
Evaluation
Success!


Structured APIs
Raw Data
DB
CSV
JSON
Preprocessing cleaning & feature engineering
Transformers & Estimators
Clean & Structured
Estimators & Models
Modeling &/or analytical techniques
Pipelines Cross Validation
Tuning
Evaluation
Success!
Evaluators Metrics

Transformer

- Tokenizer
- StopWordsRemover
- n-gram
- Binarizer
- PCA
- PolynomialExpansion
- Discrete Cosine Transform (DCT)
- StringIndexer
- IndexToString
- OneHotEncoder
- VectorIndexer
- Interaction
- Normalizer
- StandardScaler
- RobustScaler
- MinMaxScaler
- MaxAbsScaler
- Bucketizer
- ElementwiseProduct
- SQLTransformer
- VectorAssembler
- VectorSizeHint
- QuantileDiscretizer
- Imputer

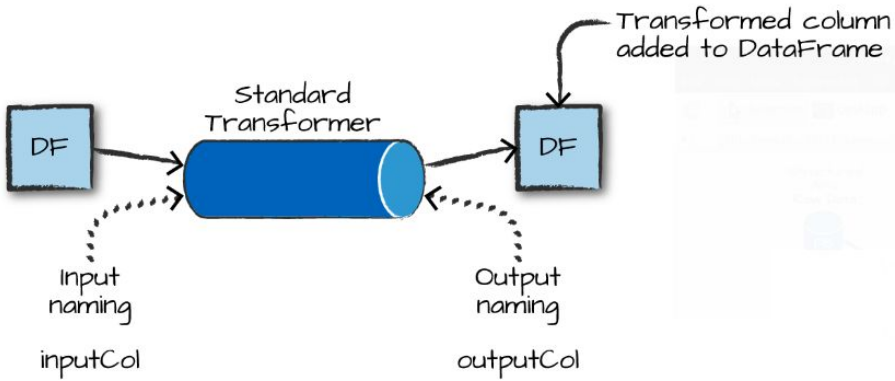


Figure 24-3. A standard transformer

Estimators

Evaluators