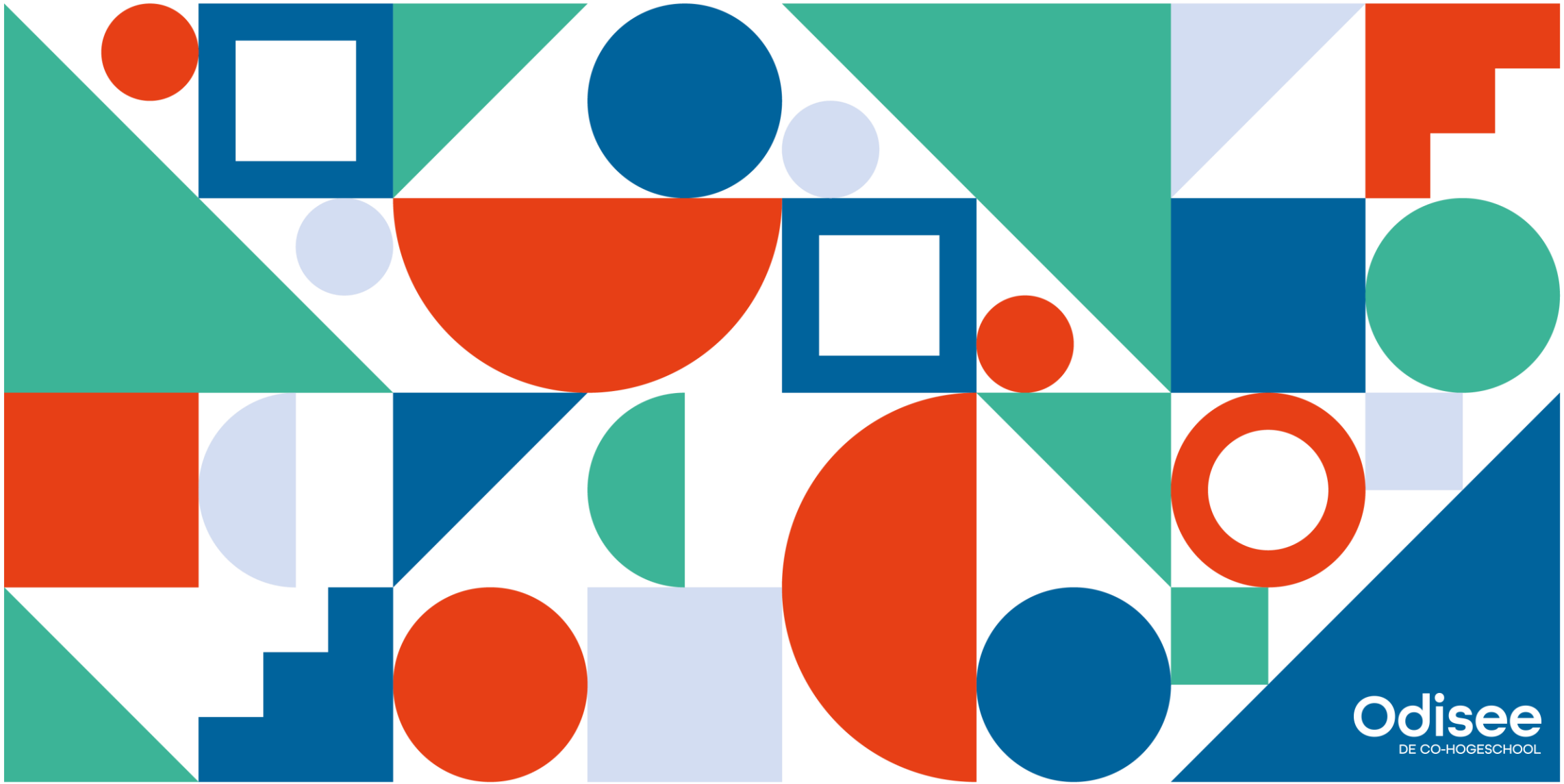

Odisee
DE CO-HOGESCHOOL

Spark – MLlib

Jens Baetens

Wat is MLlib

- Spark’s machine learning library

- Tools voor:
  - Utitilities: algebra, statistieken, data cleaning, data handling, …
  - ML technieken: classificatie, regressive, clustering, …
  - Features: extractive, transformation, dimensionality reduction, …
  - Pipelines: maken, evalueren en tuning van ML-pijplijnen
  - Persistence: bewaren en inladen van technieken, modelen, pijplijnen

MLlib

- API is gebaseerd op Spark Dataframes
  - Gebruiksvriendelijker dan RDD's
  - Sterk gelijkaardig aan sklearn
  - Laat het werken met pipelines toe
  - Wordt ook SparkML genoemd

- Een uitgebreide uitleg en voorbeeldcode vind je hier: https://spark.apache.org/docs/latest/ml-guide.html

Utilities

Data sources

- Inlezen van csv, json en andere
  - via sparkContext.read
- Image datasource
  - Laad dataset by using read.format(“image”) en lees een directory waarin de beelden staan
- Libsvm formats
  - Data reeds gesplitst in labels (doubles) en features (Vectors)

Data – pyspark.ml.linalg

- Bij sklearn werd er gewerkt met
  - Dataframes van pandas
  - Matrices en arrays van numpy

- Bij Spark werken we met
  - DataFrames van Spark
  - Matrix en Vector van Spark (gedistribueerde varianten)
    - Deze kunnen dense (alle elementen ingevuld) of sparse (sommige cellen zijn leeg) zijn

Statistieken – pyspark.ml.stat

- Summarizer – allerhande statistieken op vector
- Correlation – correlation matrix
- …

Pipelines

Onderdelen

- Dataframe
- Transformer
- Estimator
- Pipeline
- Parameter

Onderdelen

- Dataframe – bevat data
- Transformer
- Estimator
- Pipeline
- Parameter

Onderdelen

- Dataframe – bevat data
- Transformer – algoritme dat een dataframe omzet in een ander dataframe
  - ML – model zet features om naar voorspellingen
- Estimator

- Pipeline

- Parameter

Onderdelen

- Dataframe – bevat data
- Transformer – algoritme dat een dataframe omzet in een ander dataframe
  - ML – model zet features om naar voorspellingen
- Estimator – algoritme dat getrained wordt op een dataframe om een transformer te bekomen
  - De leeralgoritmes leren van de data om een model te bekomen
- Pipeline

- Parameter

Onderdelen

- Dataframe – bevat data
- Transformer – algoritme dat een dataframe omzet in een ander dataframe
  - ML – model zet features om naar voorspellingen
- Estimator – algoritme dat getrained wordt op een dataframe om een transformer te bekomen
  - De leeralgoritmes leren van de data om een model te bekomen
- Pipeline
  - Een ketting van transformers en estimators om een workflow te bekomen
- Parameter

Onderdelen

- Dataframe – bevat data
- Transformer – algoritme dat een dataframe omzet in een ander dataframe
  - ML – model zet features om naar voorspellingen
- Estimator – algoritme dat getrained wordt op een dataframe om een transformer te bekomen
  - De leeralgoritmes leren van de data om een model te bekomen
- Pipeline
  - Een ketting van transformers en estimators om een workflow te bekomen
- Parameter
  - Een gemeenschappelijke API voor transformers en estimators voor parameters in te stellen

Transformers

- Implementeert een methode .transform()
- Dataframe -> Dataframe
  - Feature transformer: een kolom (text) omzetten in een nieuwe kolom (features)
  - Learning model: een kolom (features) omzetten naar kolom (voorspellingen)

- Heeft uniek ID om parameters in te stellen

Estimators

- Implementeert een methode .fit()
- Dataframe omzetten naar een model
  - Dit model is een transformer
  - Bvb LogisticRegression is een? LogisticRegressionModel is een?
- Heeft uniek ID om parameters in te stellen

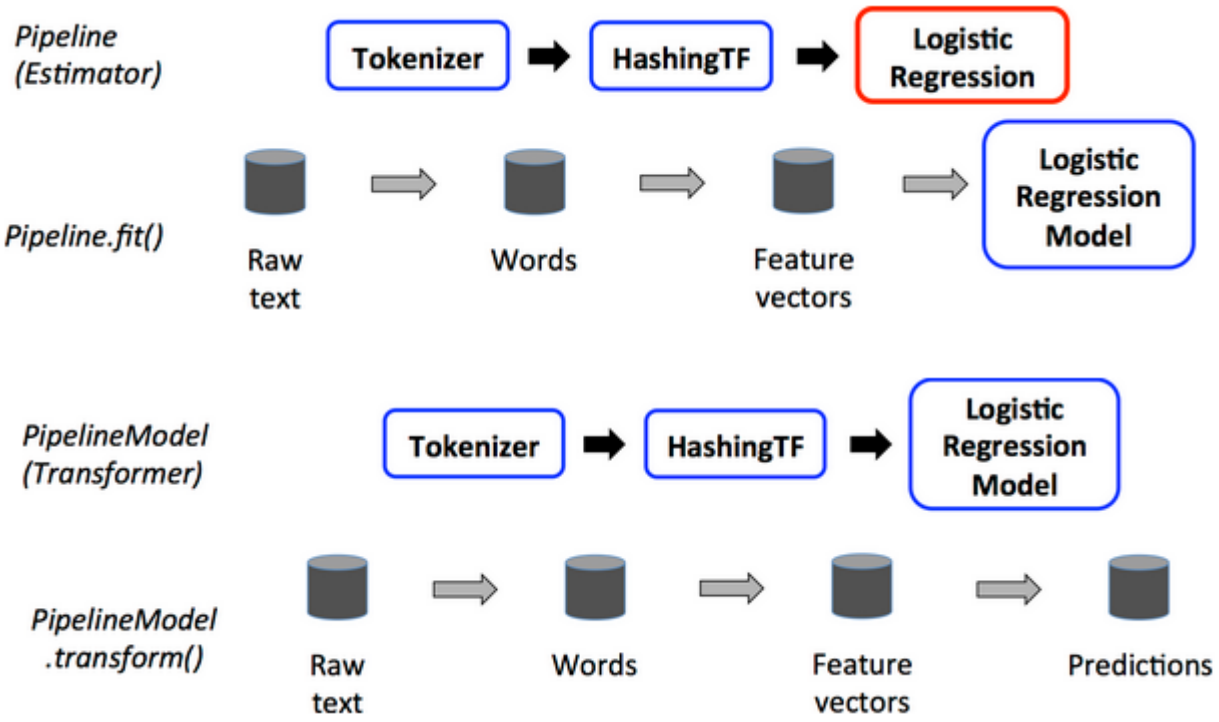
Pipelines

- Sequence of algoritmes om de workflow voor te stellen
- Verschillende stages bvb:
  - Split alle documenten in woorden
  - Zet de woorden van elk document om in feature vector
  - Train een model op basis van de feature vector

  - Is een pipeline een estimator of een transformer?

Pipelines

- Kan zowel een estimator als transformer zijn



Tuning

- CrossValidator / TrainValidationSplit
  - Estimator -> pipeline/algorithm
  - Set van ParamMaps -> parameter grid
  - Evaluator -> metriek om naar te optimaliseren

- Default is er geen parallellisatie
  - Er is een parameter om dit in te stellen
  - Pas op dat je de server niet overbelast met te hoge parallelisatie
    - Tot 10 is in de praktijk vaak geen problem (op echte clusters)

Cross - validation

- Analoog aan GridSearchCV

- Aantal folds in te stellen
  - K=1 is speciale variant -> TrainTestValidation
- Evaluatie door
  - RegressionEvaluator
  - BinaryClassificationEvaluator
  - MulticlassClassificationEvaluator
  - MultilabelClassificationEvaluator

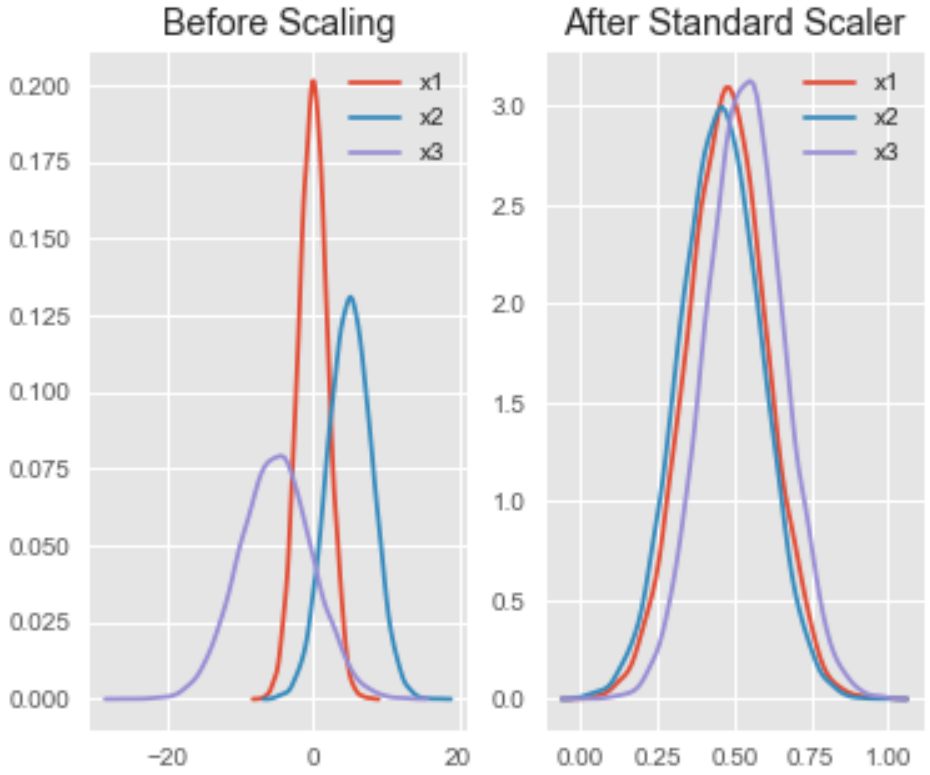
Features

Feature transformers

- Tokenizer
- StopWordsRemover
- n –gram
- Binarizer
- PCA
- PolynomialExpansion
- StringIndexer (Ordinal Encoding)
- IndexToString (Reverse)
- OneHotEncoder
- ElementwiseProduct
- Interaction (Combinaties)
- Bucketizer
- SQLTransformer
- VectorAssembler
- VectorSizeHint
- QuantileDiscretizer

Scalers

- Normalizer
- StandardScaler
- RobustScaler
- MinMaxScaler
- MaxAbsScaler



Imputers

- Imputer

Feature extractors

- Convert text to feature vector or reduce dimensions
  - Hashing TF (term frequency)
  - Word2Vec
  - CountVectorizer (term frequency)
  - FeatureHasher

- Hashing = waarde omzetten naar andere waarde
  - Kan niet omgekeerd worden
  - Pas op voor hashing conflicten

Feature Selectors

- Manier om een deel van de features te kiezen
  - zelf of op basis van een algoritme

- VectorSlicer
- Rformula
- ChiSqSelector
- UnivariateFeatureSelector
- VarianceThresholdSelector

ML - technieken

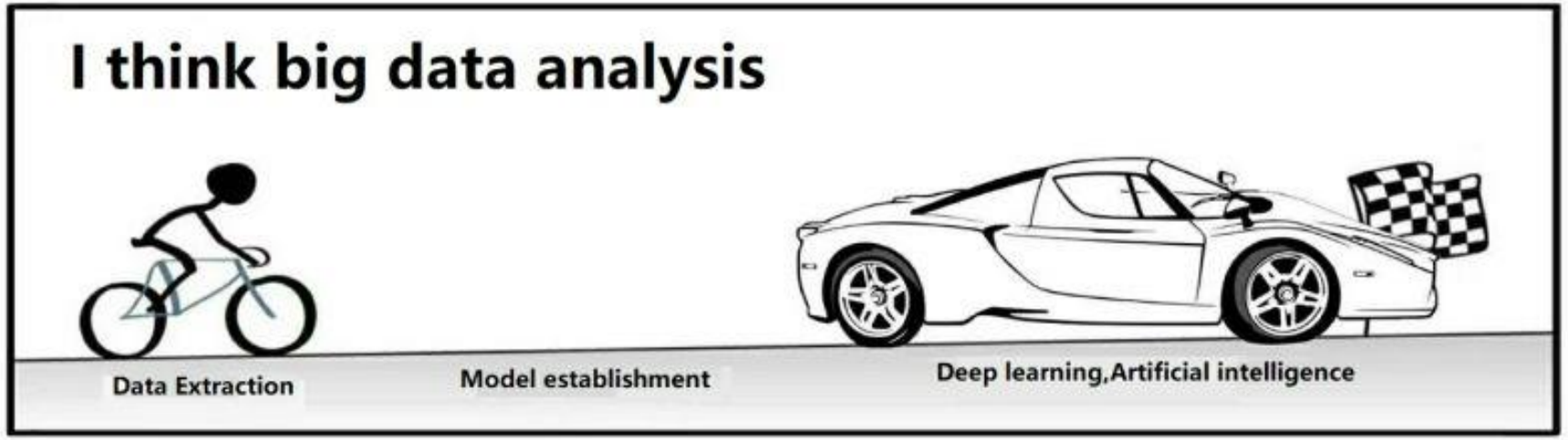
Technieken

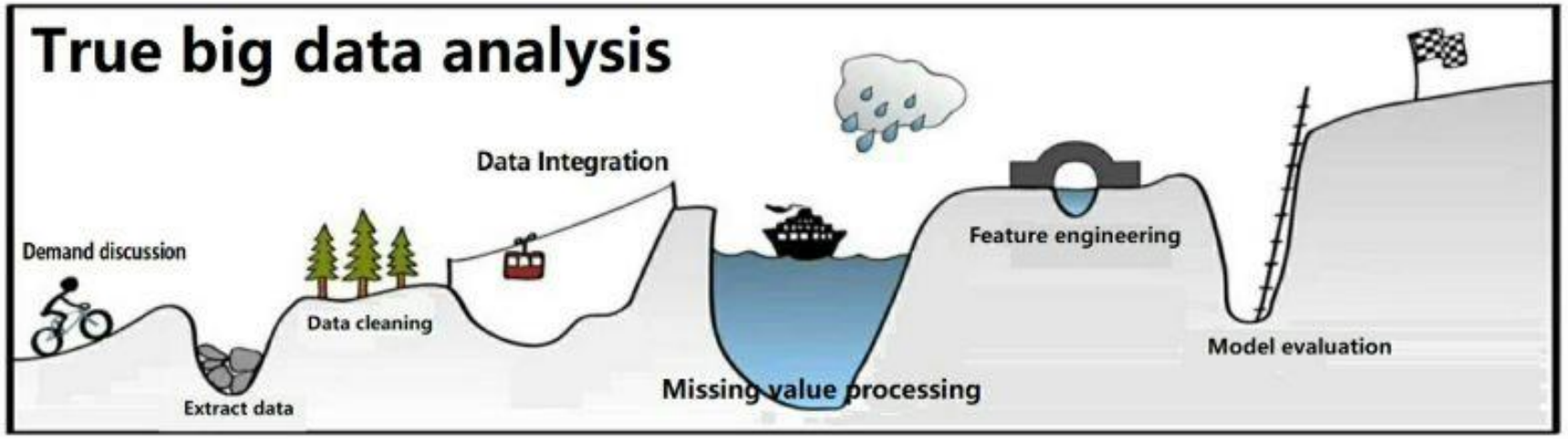
- Classificatie/Regressie
  - http://spark.apache.org/docs/latest/ml-classification-regression.html#classification
- Clustering
  - http://spark.apache.org/docs/latest/ml-clustering.html

Persistence

Opslaan en inladen van modellen

- Bewaar een model/pipeline op de cluster
- Elk model/pipeline heeft een read()/write() functie
  - https://spark.apache.org/docs/2.4.0/api/python/pyspark.ml.html#pyspark.ml.util.MLReader
  - https://spark.apache.org/docs/2.4.0/api/python/pyspark.ml.html#pyspark.ml.util.MLWriter


I think big data analysis
Data Extraction
Model establishment
Deep learning,Artificial intelligence


True big data analysis
Demand discussion
Extract data
Data cleaning
Data Integration
Missing value processing
Feature engineering
Model evaluation