everis
an NTT DATA Company

素晴らしいガイド

FROM A PYTHON APPLICATION TO A MACHINE LEARNING PREDICTION WITH SPARK

MAY 2020

TECH THURSDAY

1

About the Demo

2

Recommendation Pipeline

3

Prediction Pipeline

4

REST API on Top of Spark ML Models

ABOUT THIS TALK

Why this talk?

This talk has as objective to promote and engage everis Technology people to start getting more understanding in:

PYTHON

Foment and promote Python solutions and applications.

+

MACHINE LEARNING

To be more familiar with this techniques and kind of solutions and frameworks.

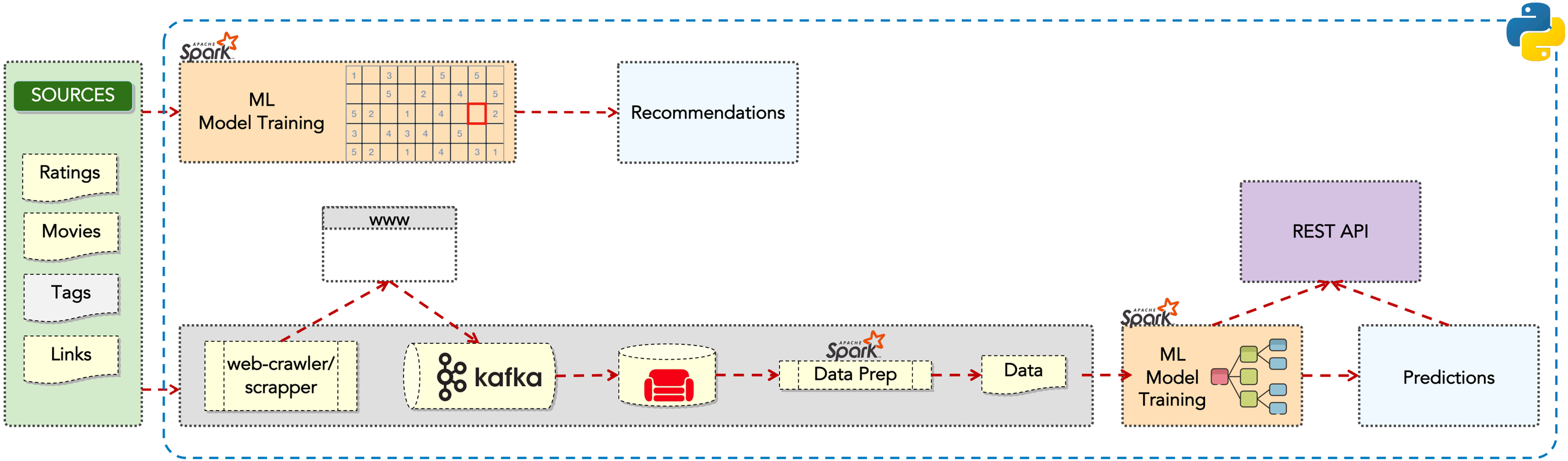
=

PROFICIENCY

To get more these professional skills within the company for

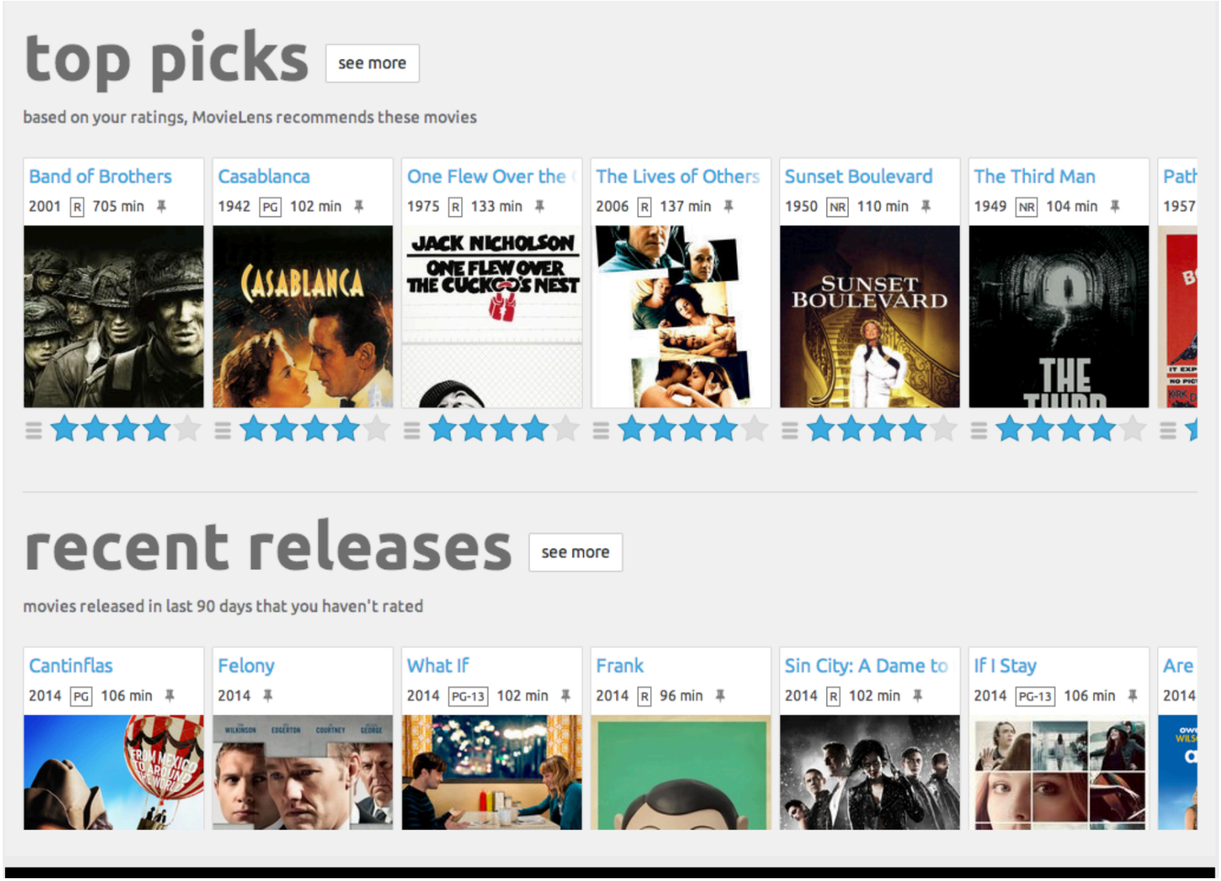
WHAT WE WILL SEE

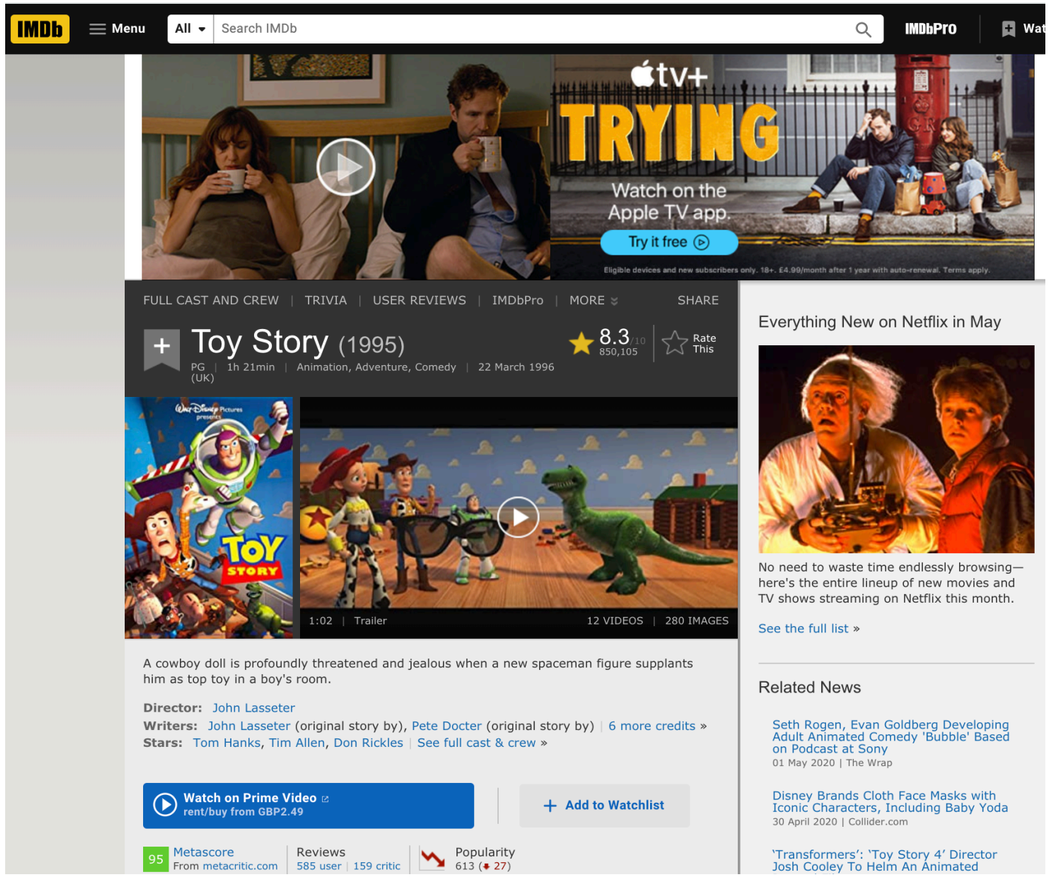
Why this talk?



ABOUT THE DATA

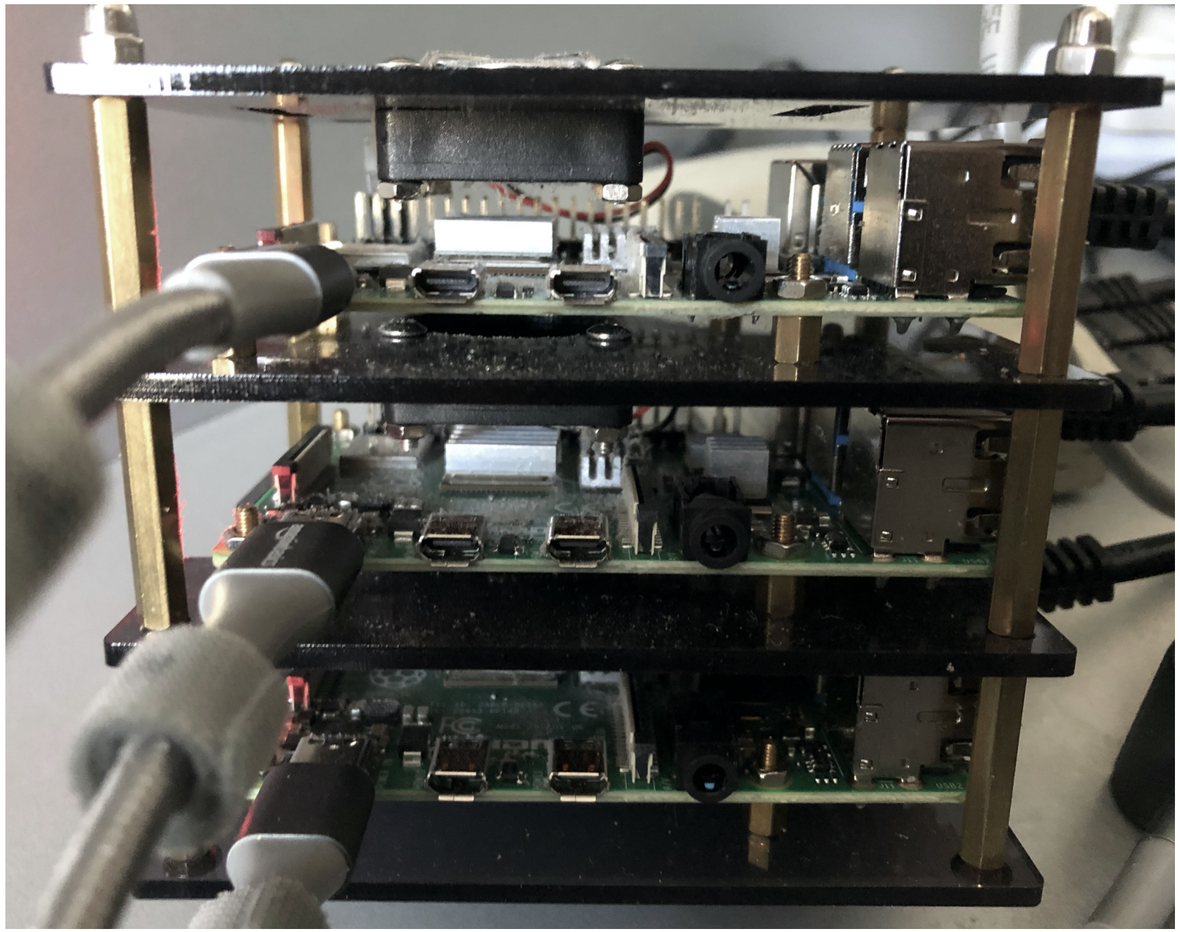
From where these data came from

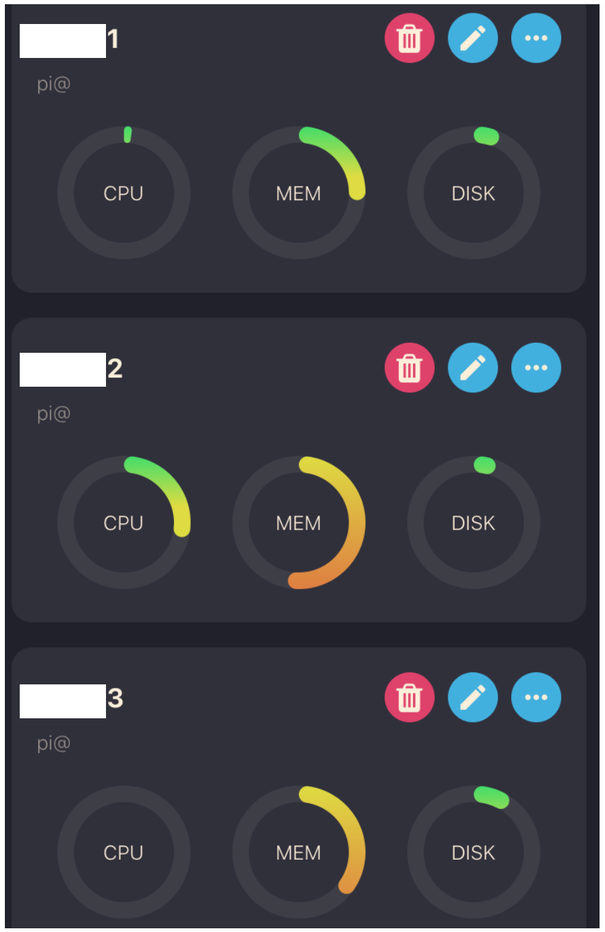




THE USED CLUSTER

Raspberry Pi Cluster

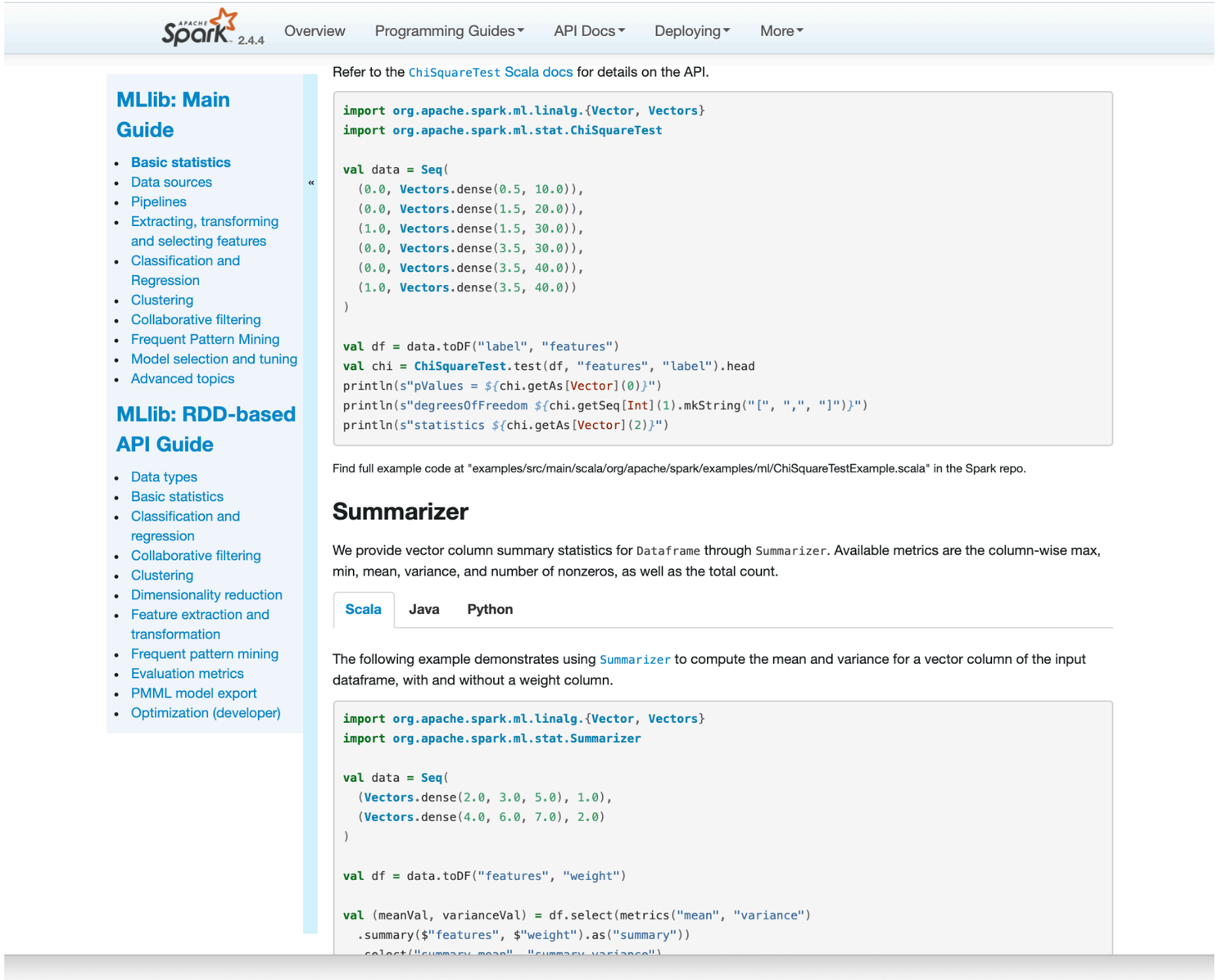




Apache Spark
kafka
Apache CouchDB relax
hadoop

SPARK ML

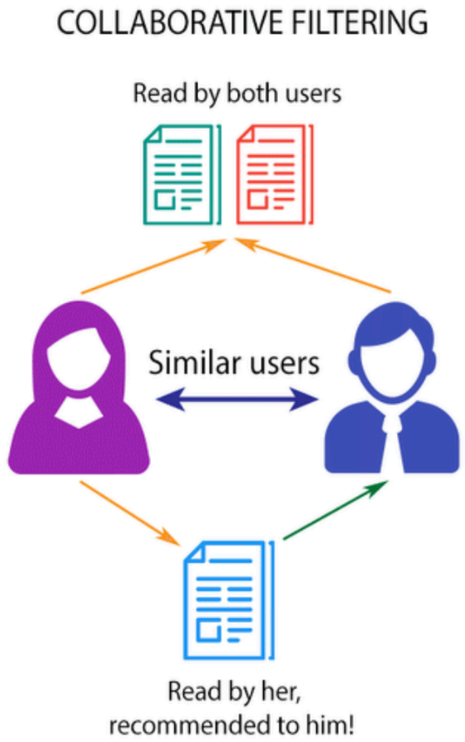
Version 2.4.4

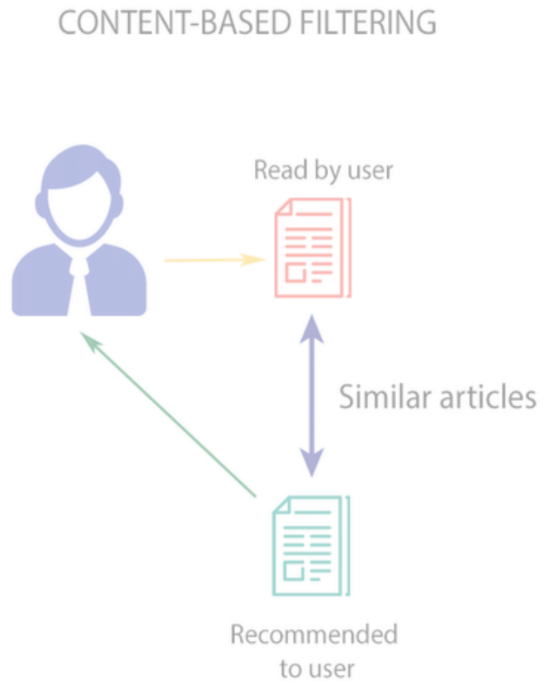


Recommendation Pipeline

A LITTLE BIT ABOUT RECOMMENDATION SYSTEMS

Recommendation References:



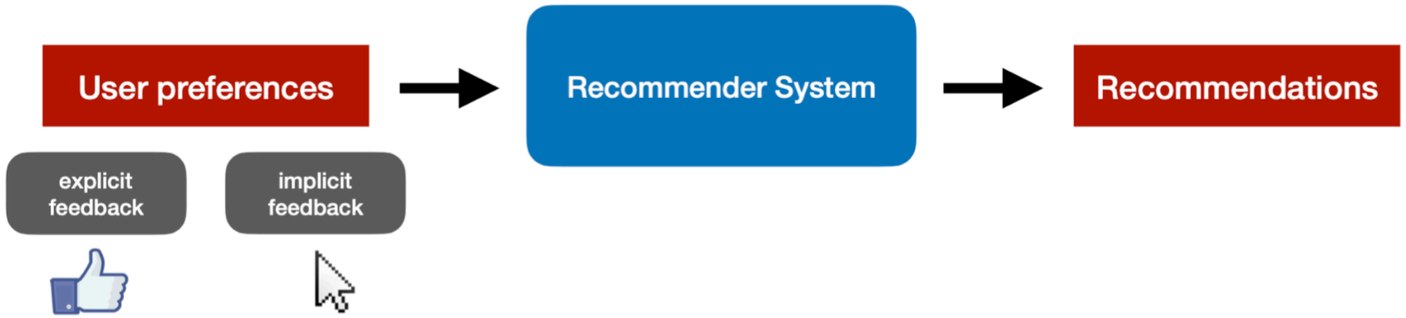


Spark ML: Alternating Least Square (ALS)

Source: A

A LITTLE BIT ABOUT RECOMMENDATION SYSTEMS

How it works?



Collaborative filtering

user \ item									
John	1		3			5		5	
Erica			5		2		4		5
Anne	5	2		1		4			2
Liz	3		4	3	4		5		
Jim	5	2		1		4		3	1

Content-based filtering

Source: A

A LITTLE BIT ABOUT RECOMMENDATION SYSTEMS

Collaborative Filtering: Factorization Matrix

	Item1	Item2	Item3	Item4
User1	x	x		x
User2		x	x	
User3				x
User4	x	x	x	

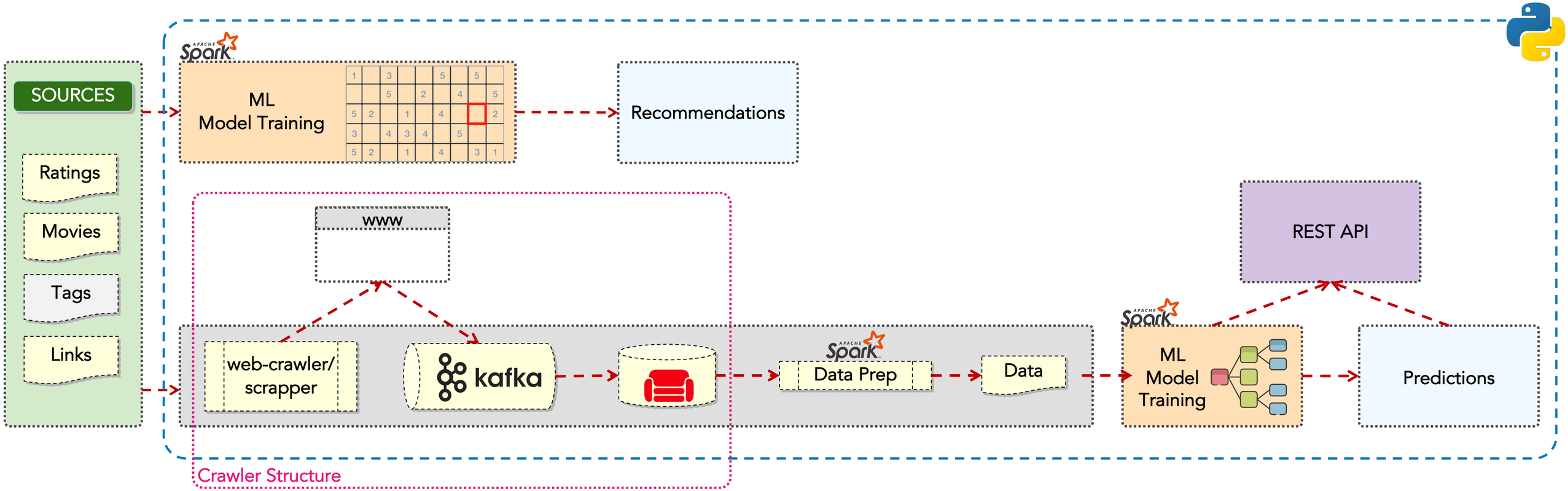
Let’s see an example with some theory

Source: A

Prediction Pipeline

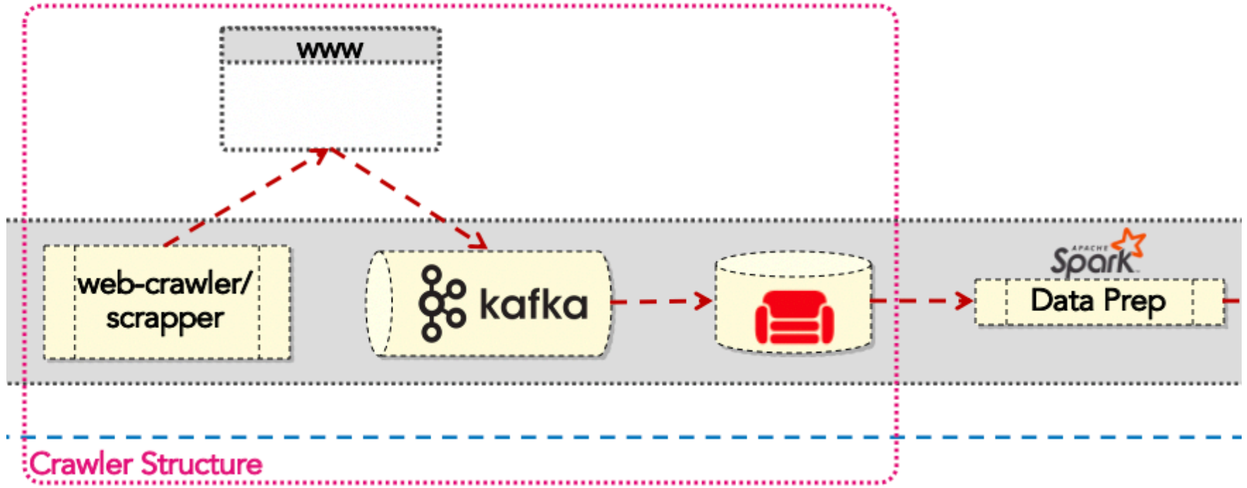
LET'S LOOK IT AGAIN

The prediction pipeline



THE CRAWLER STRUCTURE AND THE DATA PREPARATION

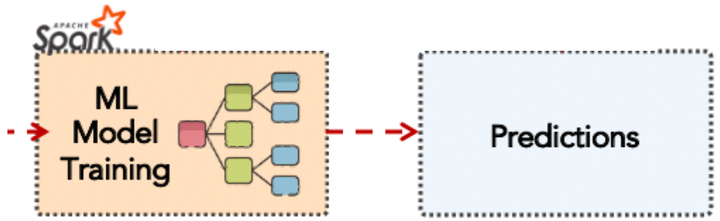
Understanding the process: The objective is to get more data about the movies.



Let’s take a look on it!

PREDICTION MODEL

Predict the budget value for each movie based on the Director, Genre and Rating



Spark 2.4.4 Overview Programming Guides API Docs Deploying

MLlib: Main Guide

- Basic statistics
- Data sources
- Pipelines
- Extracting, transforming and selecting features
- **Classification and Regression**
- Clustering
- Collaborative filtering
- Frequent Pattern Mining
- Model selection and tuning
- Advanced topics

MLlib: RDD-based API Guide

- Data types
- Basic statistics
- Classification and

- Regression
  - Linear regression
  - Generalized linear regression
    - Available families
  - Decision tree regression
  - Random forest regression
  - Gradient-boosted tree regression
  - Survival regression
  - Isotonic regression
- Linear methods
- Decision trees
  - Inputs and Outputs
    - Input Columns
    - Output Columns
- Tree Ensembles
  - Random Forests
    - Inputs and Outputs
      - Input Columns
      - Output Columns (Predictions)
  - Gradient-Boosted Trees (GBTs)
    - Inputs and Outputs
      - Input Columns
      - Output Columns (Predictions)

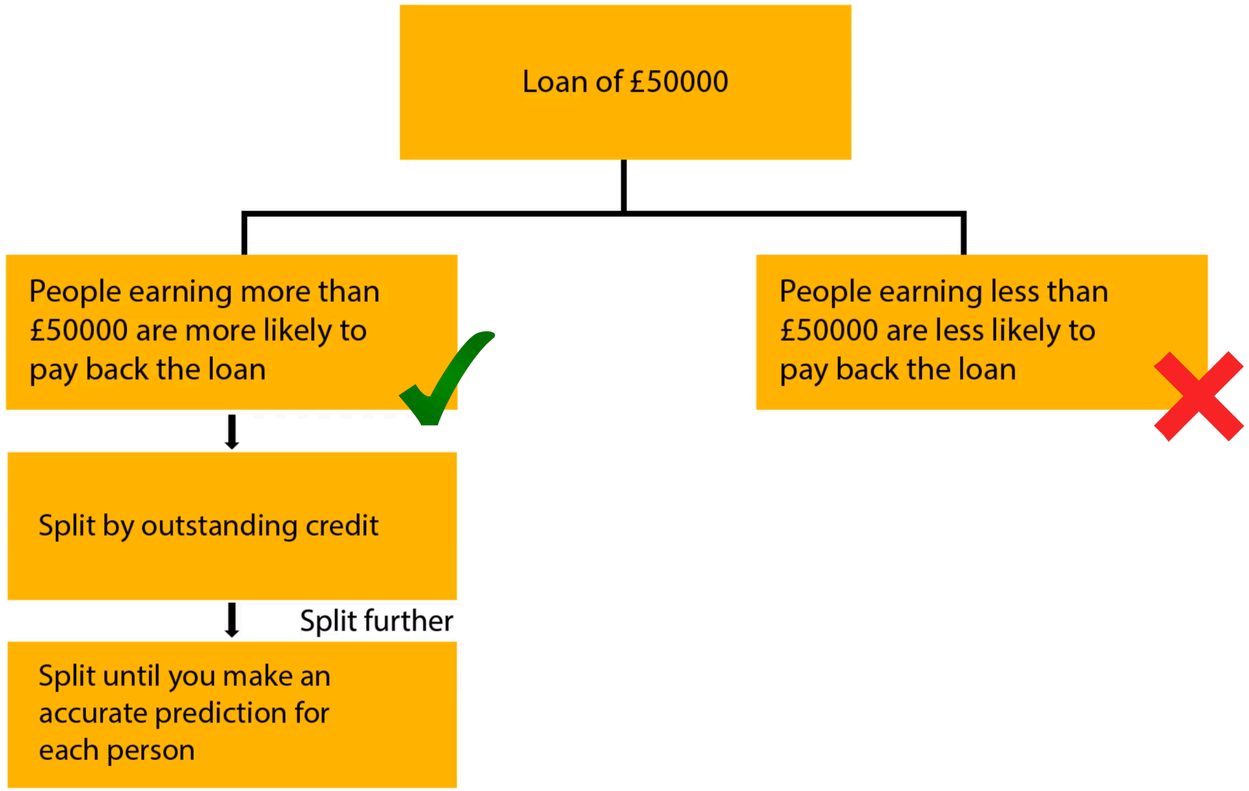
PREDICTION MODEL

A little bit about the model theory

Decision Tree

- It starts with all the data and then it tries to find the best split up the individuals such that we've got a better prediction for both groups.
- It can be used to predict a number or to predict the probability of an event.
- Decision Tree models are very sensible if there is/are update(s) in the Analytical Base Table (ABT).
- Decision Tree Example: http://www.r2d3.us/visual-intro-to-machine-learning-part-1/

Example: bank loan repayment



Source: C

PREDICTION MODEL

A little bit about the model theory

Random Forest

- Split data into thousands of random subsamples.
- Build a simple decision tree, for example tree leaves deep.
- Obtain a thousand each with a different prediction.
- Average all the decisions.
- The average prediction of many simple trees is more accurate than the single prediction of a very complicated tree.

Gradient Boosted

- It follows same Random Forest concept, but it applies some weights “scoring”.
- Combine trees at the beginning, instead of the end.
- Build one tree at time.

Source: C

SPARK FEATURE EXTRACTION

Continuous Features Transformers

- Standard Scaler
- Min/Max Scaler
- Binarizer
- Bucketizer
- ...

Categorical Features Transformers

- StringIndexer
- One-HotEncoder
- …

NLP Features Transformers

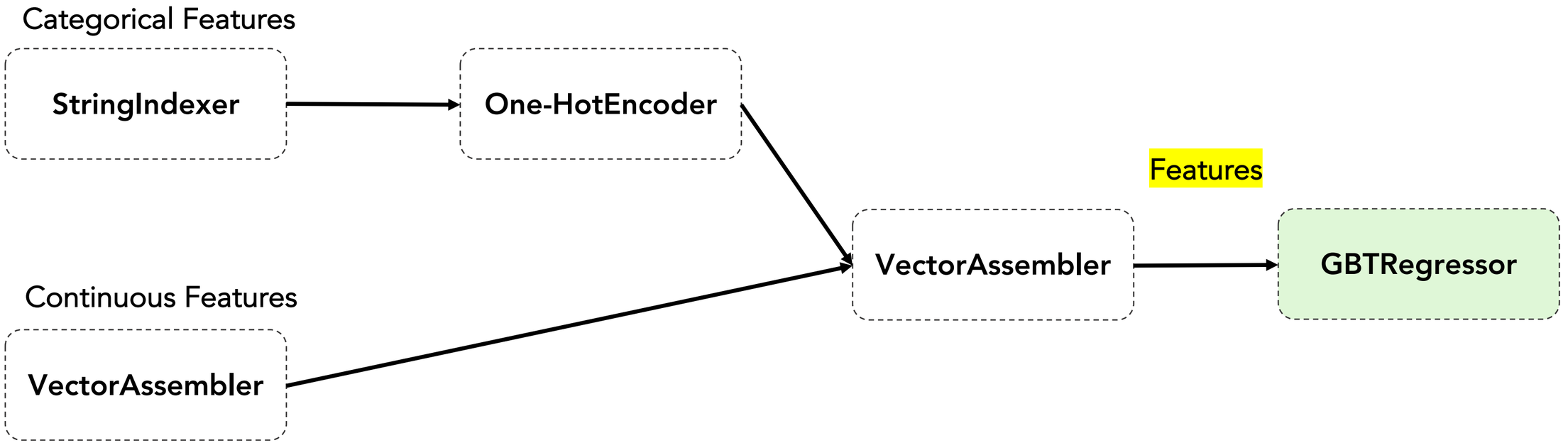
- NGram
- Word2Vec
- Hash Term Frequency
- …

All about Feature Extraction:

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

USED SPARK PREDICTION PIPELINE

What are the steps?



Let's get out of here and see some code, please!

素晴らしいガイド

REST API on Top of Spark ML Models

NTT DATA

everis
an NTT DATA Company

SPARK MODELS AS A REST API

Deploying in Production

Using Spark

- Through Spark, Streaming or JobServer (with Livy) where the prediction is performed for each micro-batch or active spark context.

Without Spark

These solutions has as objective to deploy spark models to be used outside Spark without a SparkContext

- **PMML (Predictive Model Markup Language):** Supported initially in the Mllib with RDD or using a Thrird-Party component.
- **MLeap:** It's becoming the most applied solution and recommended also by Databricks:
  - https://docs.databricks.com/applications/machine-learning/model-export-import/mleap-model-export.html#export-and-import-models-in-python
  - https://docs.databricks.com/applications/mlflow/mleap-model-deployment-on-sagemaker.html

Source: E,F,G

REFERENCES

A. Recommendation Systems: https://github.com/maxhumber/BRE

B. Spark ALS Parameters:

1. https://spark.apache.org/docs/latest/api/scala/index.html#org.apache.spark.ml.recommendation.ALS
2. https://spark.apache.org/docs/2.4.4/ml-collaborative-filtering.html#collaborative-filtering

C. Business Analytics: Decision Making Using Data Course – University Cambridge Judge Business School

D. Feature Extraction: https://spark.apache.org/docs/2.4.4/ml-features.html

E. Spark PMML: https://spark.apache.org/docs/2.3.0/mllib-pmml-model-export.html

F. JPMML-Evaluator-Spark: https://github.com/jpmml/jpmml-evaluator-spark

G. Mleap Documentation: https://mleap-docs.combust.ml/

H. Mleap Git Repo: https://github.com/combust/mleap

I. Spark API: https://www.codementor.io/@jadianes/building-a-web-service-with-apache-spark-flask-example-app-part2-du1083854

That’s all folks!