HITACHI
Inspire the Next

Pentaho – Simplifier le Machine Learning avec Pentaho Machine Intelligence

Jean-François Monteil
Ingénieur Avant-Vente/ IoT et Big Data Analytics


2 2

Chart of the Week

THE EXTRAORDINARY SIZE OF AMAZON IN ONE CHART

Amazon is bigger than most brick and mortar retailers put together

Market value as of December 30, 2016

Here is how the value of these companies has changed over the last 10 years:

COMPANY	MARKET VALUE 2006	MARKET VALUE 2016	% CHANGE
sears	$27.8B	$1.1B	↘ 96%
JCPenney	$18.1B	$2.6B	↘ 86%
NORDSTROM	$12.4B	$8.3B	↘ 33%
KOHL'S	$24.2B	$8.8B	↘ 64%
macy's	$24.2B	$11.0B	↘ 55%
BEST BUY	$28.4B	$13.2B	↘ 54%
TARGET	$51.3B	$40.6B	↘ 21%
Walmart	$214.0B	$212.4B	↘ 1%
amazon	$17.5B	$355.9B	↗ 1,934%

Source: Yahoo Finance (peak value in 2006), Google Finance (values for Dec 30, 2016)

visualcapitalist.com


Google
amazon.com
Walmart
WHOLE FOODS MARKET

Agenda

HITACHI
Inspire the Next

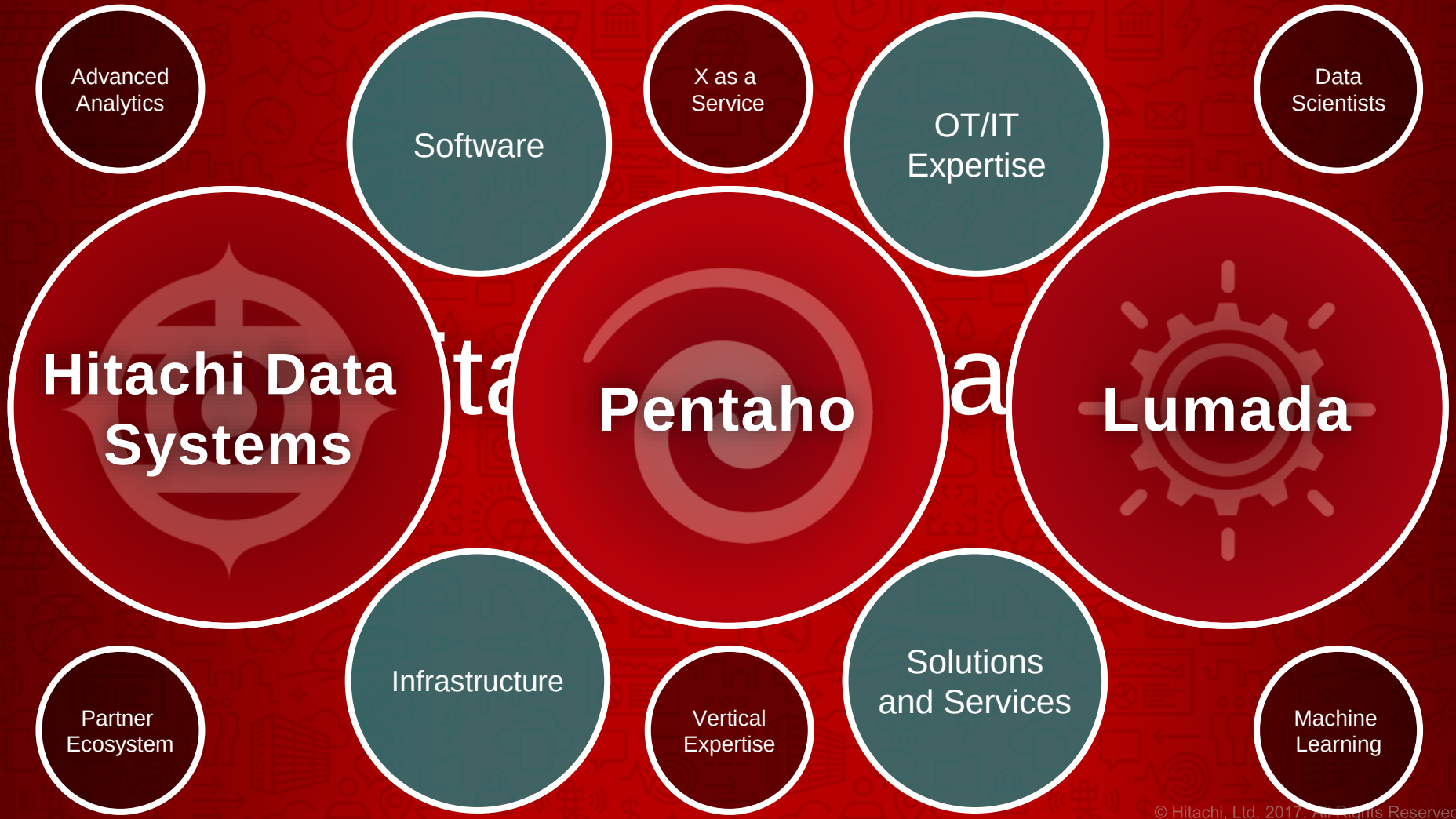
- Présentation de Pentaho dans l'ecosystème Hitachi Vantara
- Les 3 Challenges du Machine Learning
- Roadmap Pentaho V8.1

© Hitachi Vantara Corporation 2018. All Rights Reserved

HITACHI
Inspire the Next

Hitachi Vantara

© Hitachi Vantara Corporation 2018. All Rights Reserved



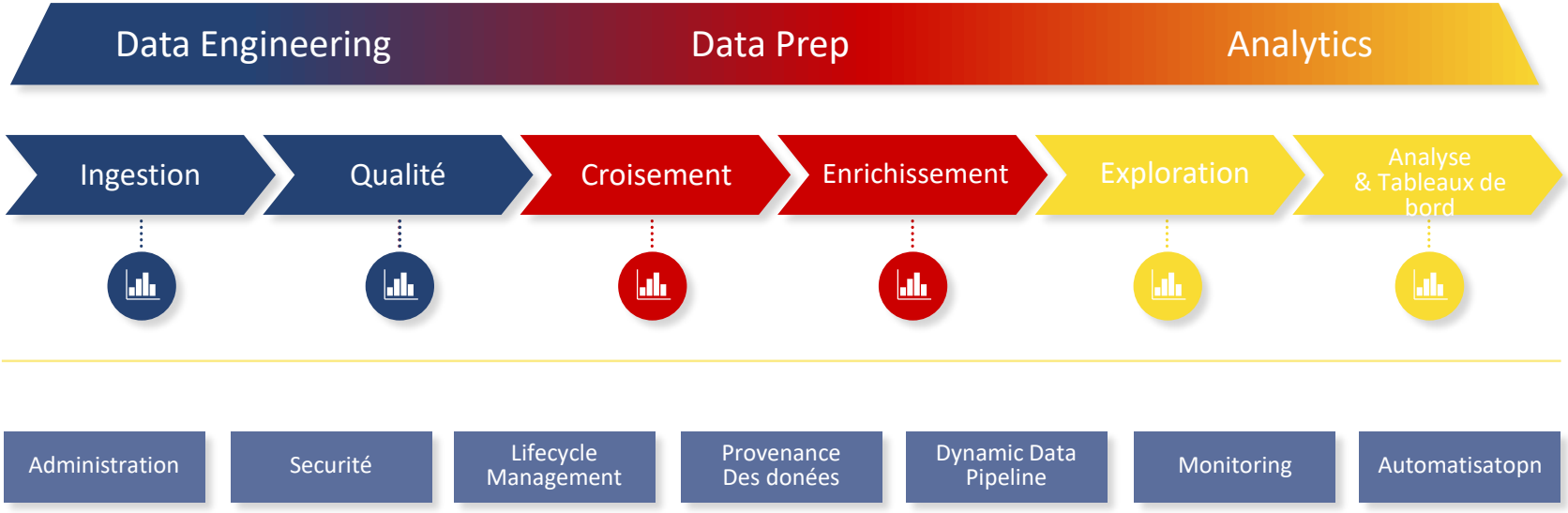
© Hitachi, Ltd. 2017. All Rights Reserved

La valeur d'Hitachi

HITACHI
Inspire the Next

© Hitachi, Ltd. 2017. All Rights Reserved

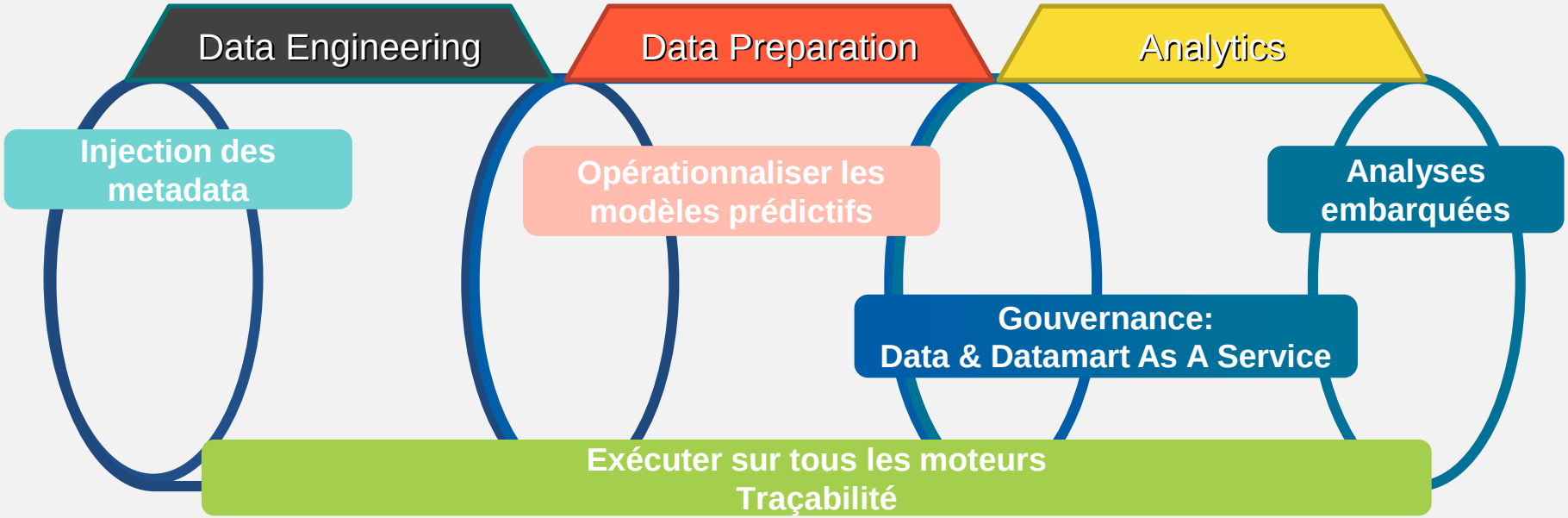
Data Analytics : A Single Flow



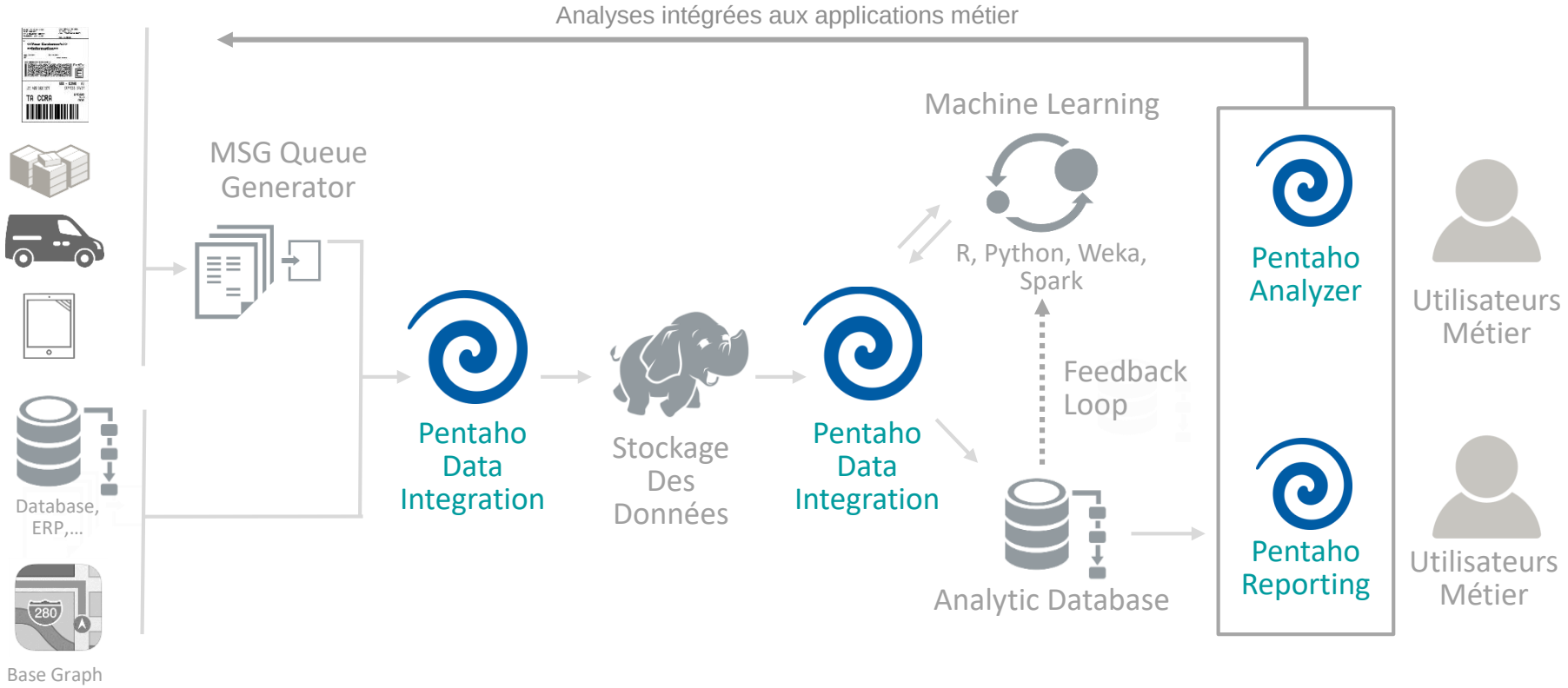
© Hitachi Vantara Corporation 2018. All Rights Reserved

Nos points clefs

HITACHI
Inspire the Next



Architecture type

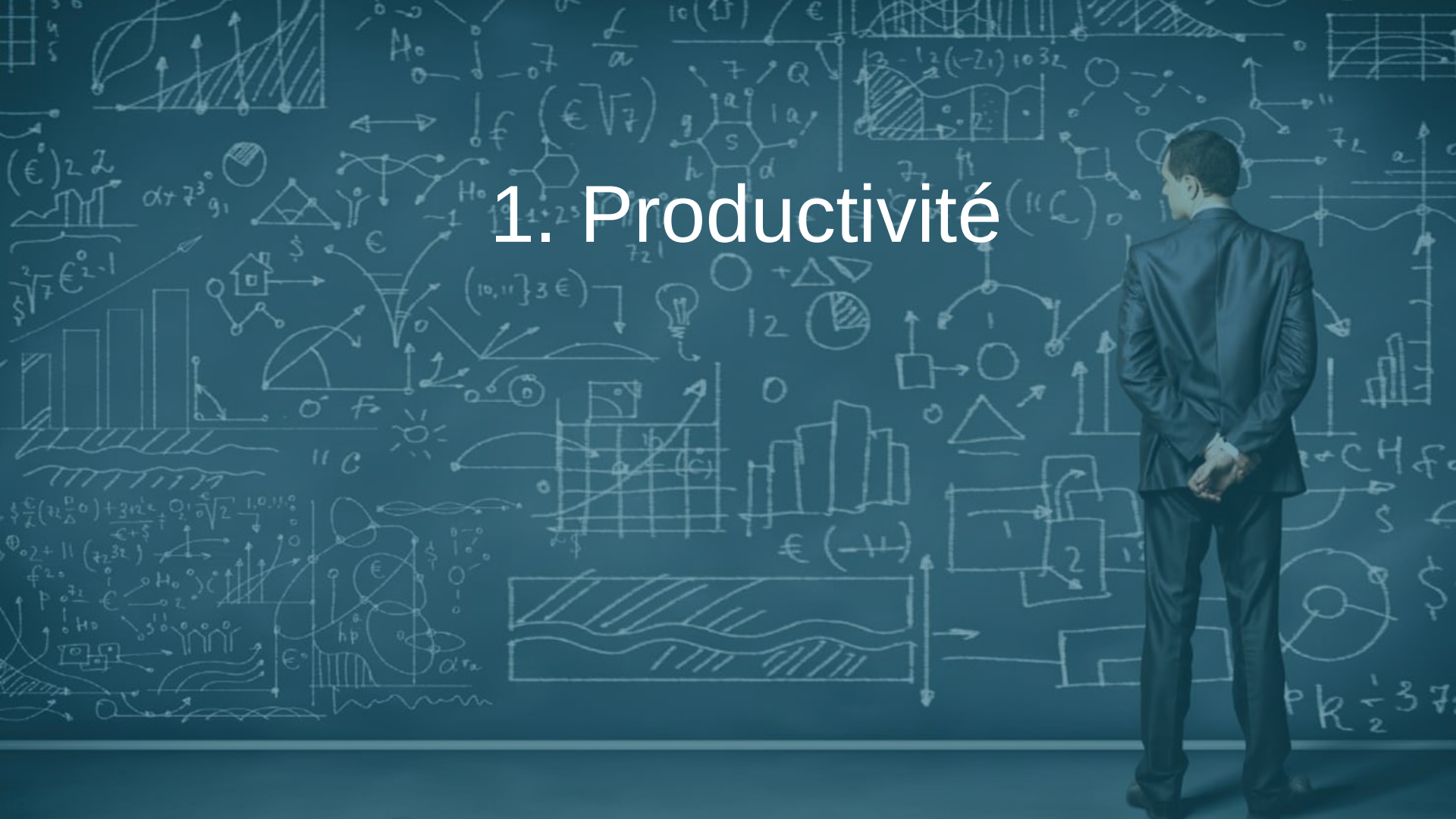


© Hitachi Vantara Corporation 2018. All Rights Reserved

HITACHI
Inspire the Next

Les 3 challenges du machine learning

© Hitachi Vantara Corporation 2018. All Rights Reserved

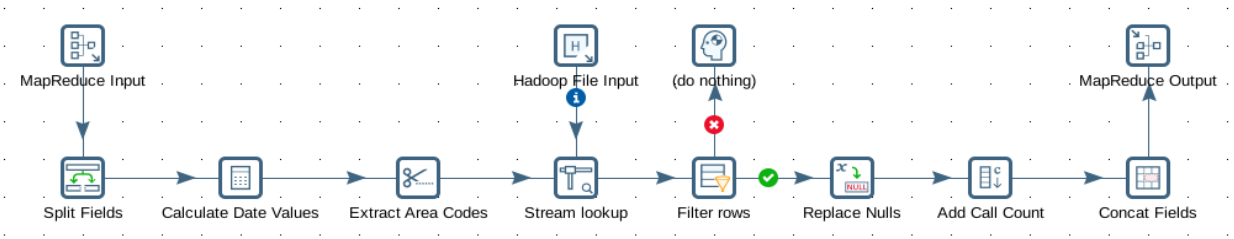

1. Productivité

Simplifier la preparation des données

HITACHI
Inspire the Next

Exemple de flux de donnée Pentaho 0 code

GUI Easy



Code équivalent

Java/ Code Equivalent

```
import java.sql.*;
import javax.sql.*;
import javax.ejb.*;
import javax.naming.*;

public class ConnectionPoolingBean implements SessionBean {

    // ...

    public void ejbCreate() throws CreateException {
        ctx = new InitialContext();
        ds = (DataSource)ctx.lookup("jdbc/fastCoffeeDB");
    }

    public void updatePrice(float price, String coffeeName,
                            String username, String password)
        throws SQLException{

        Connection con;
        PreparedStatement pstmt;
        try {
            con = ds.getConnection(username, password);
            con.setAutoCommit(false);
            pstmt = con.prepareStatement("UPDATE COFFEES " +
                                         "SET PRICE = ? " +
                                         "WHERE COF_NAME = ?");
            pstmt.setFloat(1, price);
            pstmt.setString(2, coffeeName);
            pstmt.executeUpdate();

            con.commit();
            pstmt.close();

        } finally {
            if (con != null) con.close();
        }
    }
```

```
public static class TokenizerMapper
     extends Mapper<Object, Text, Text, IntWritable>{

  static enum CountersEnum { INPUT_WORDS }

  private final static IntWritable one = new IntWritable(1);
  private Text word = new Text();

  private boolean caseSensitive;
  private Set<String> patternsToSkip = new HashSet<String>();

  private Configuration conf;
  private BufferedReader fis;

  @Override
  public void setup(Context context) throws IOException,
      InterruptedException {
    conf = context.getConfiguration();
    caseSensitive = conf.getBoolean("wordcount.case.sensitive", true);
    if (conf.getBoolean("wordcount.skip.patterns", true)) {
      URI[] patternsURIs = Job.getInstance(conf).getCacheFiles();
      for (URI patternsURI : patternsURIs) {
        Path patternsPath = new Path(patternsURI.getPath());
        String patternsFileName = patternsPath.getName().toString();
        parseSkipFile(patternsFileName);
      }
    }
  }
```

```
private void parseSkipFile(String fileName) {
  try {
    fis = new BufferedReader(new FileReader(fileName));
    String pattern = null;
    while ((pattern = fis.readLine()) != null) {
      patternsToSkip.add(pattern);
    }
  } catch (IOException ioe) {
    System.err.println("Caught exception while parsing the cached file '"
        + StringUtils.stringifyException(ioe));
  }
}

@Override
public void map(Object key, Text value, Context context
                ) throws IOException, InterruptedException {
  String line = (caseSensitive) ?
      value.toString() : value.toString().toLowerCase();
  for (String pattern : patternsToSkip) {
    line = line.replaceAll(pattern, "");
  }
  StringTokenizer itr = new StringTokenizer(line);
  while (itr.hasMoreTokens()) {
    word.set(itr.nextToken());
    context.write(word, one);
    Counter counter = context.getCounter(CountersEnum.class.getName(),
        CountersEnum.INPUT_WORDS.toString());
    counter.increment(1);
  }
}
```

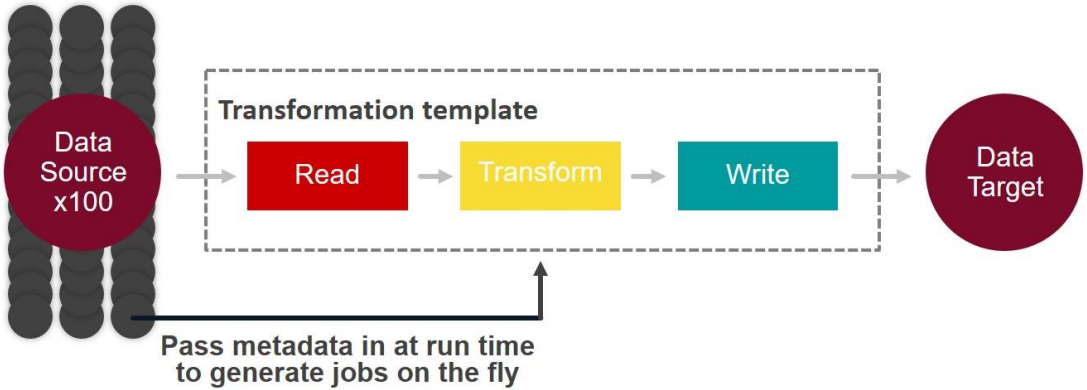
```
public static void main(String[] args) throws Exception {
  Configuration conf = new Configuration();
  GenericOptionsParser optionParser = new GenericOptionsParser(conf, args);
  String[] remainingArgs = optionParser.getRemainingArgs();
  if ((remainingArgs.length != 2) && (remainingArgs.length != 4)) {
    System.err.println("Usage: wordcount <in> <out> [-skip skipPatternFile]");
    System.exit(2);
  }
  Job job = Job.getInstance(conf, "word count");
  job.setJarByClass(WordCount2.class);
  job.setMapperClass(TokenizerMapper.class);
  job.setCombinerClass(IntSumReducer.class);
  job.setReducerClass(IntSumReducer.class);
  job.setOutputKeyClass(Text.class);
  job.setOutputValueClass(IntWritable.class);

  List<String> otherArgs = new ArrayList<String>();
  for (int i=0; i < remainingArgs.length; ++i) {
    if ("-skip".equals(remainingArgs[i])) {
      job.addCacheFile(new Path(remainingArgs[++i]).toUri());
      job.getConfiguration().setBoolean("wordcount.skip.patterns", true);
    } else {
      otherArgs.add(remainingArgs[i]);
    }
  }
  FileInputFormat.addInputPath(job, new Path(otherArgs.get(0)));
  FileOutputFormat.setOutputPath(job, new Path(otherArgs.get(1)));

  System.exit(job.waitForCompletion(true) ? 0 : 1);
}
```


© Hitachi Vantara Corporation 2018. All Rights Reserved

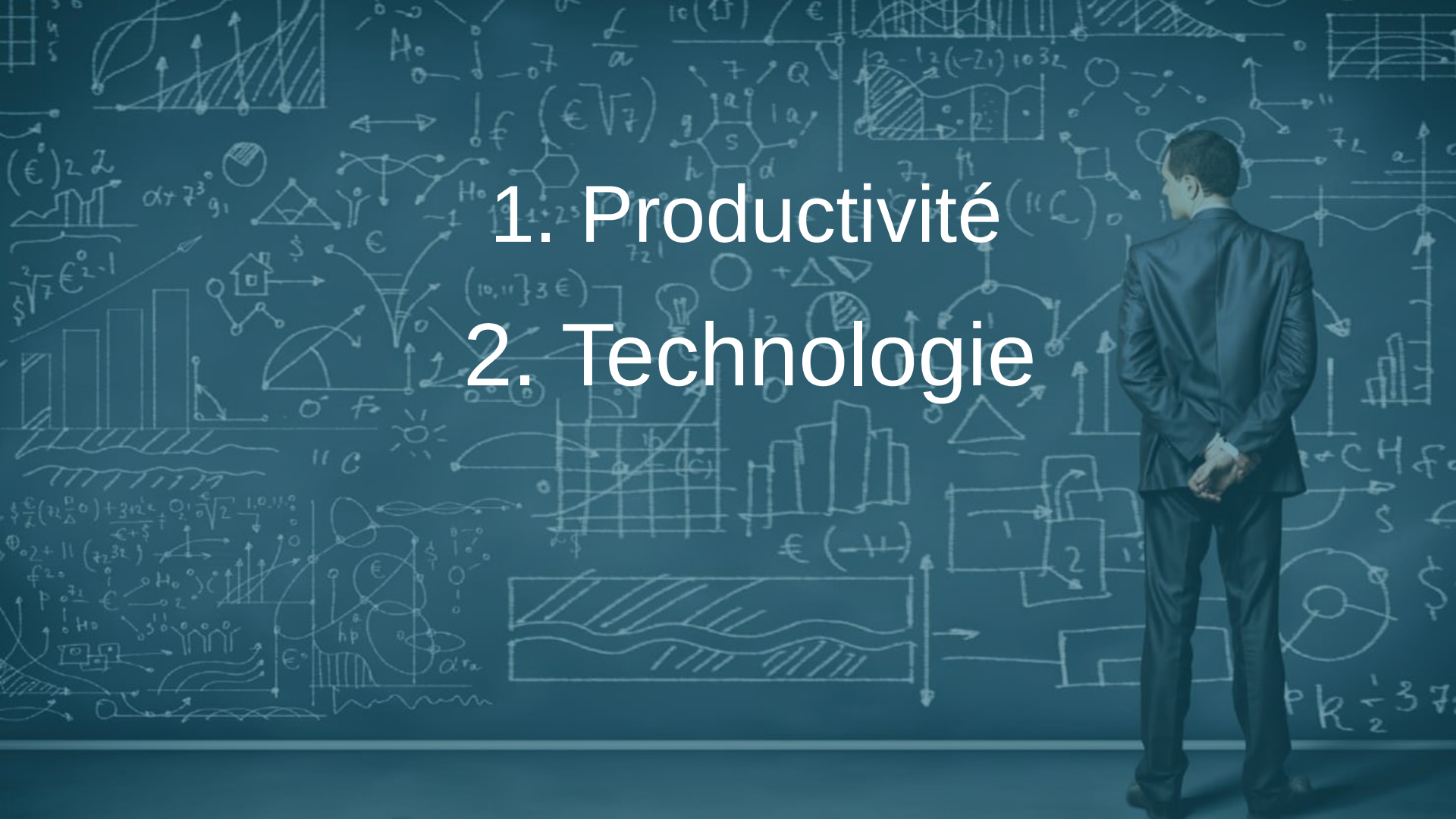
Metadata Injection

HITACHI
Inspire the Next

- Approche “template” centralisée
  - Reduction des coûts de développement
  - Centralisation des règles métier
  - Rationalisation de la maintenance



© Hitachi Vantara Corporation 2018. All Rights Reserved


1. Productivité
2. Technologie

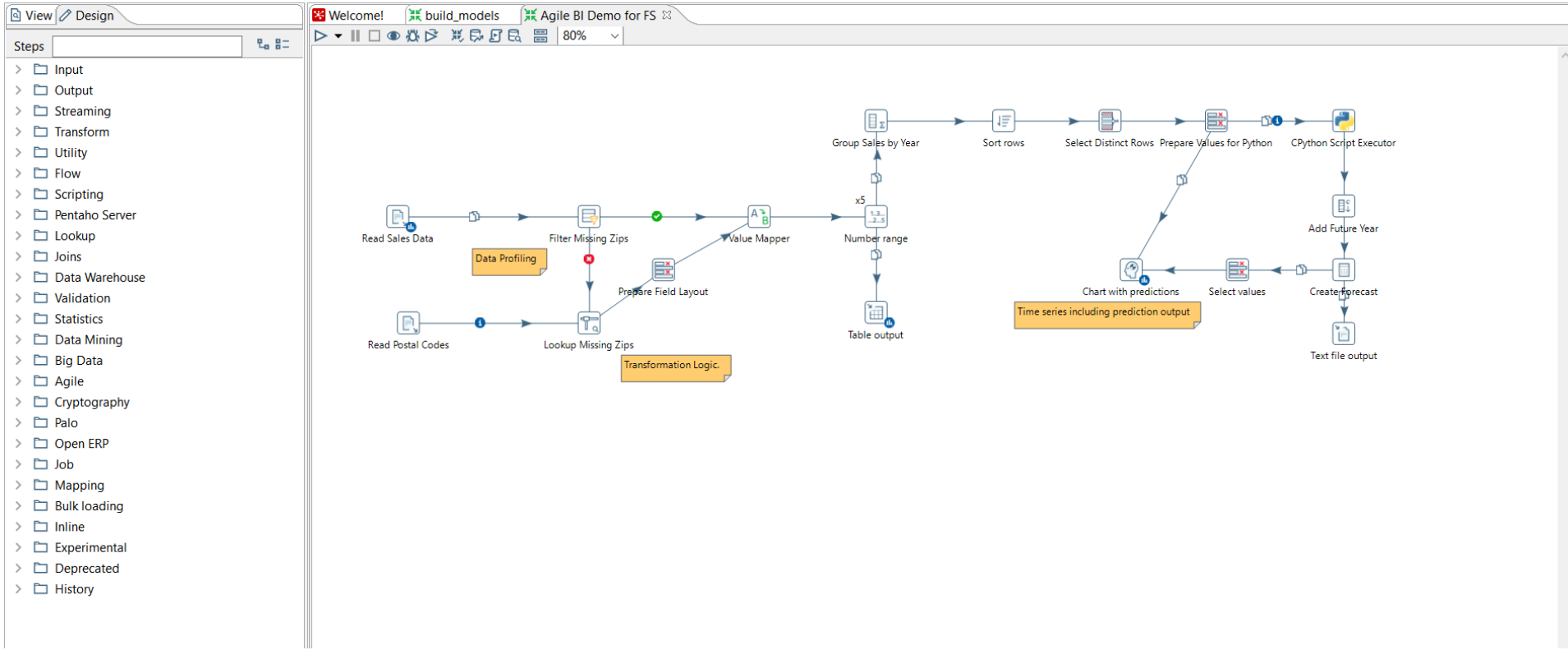
Technologies (langages)

HITACHI
Inspire the Next

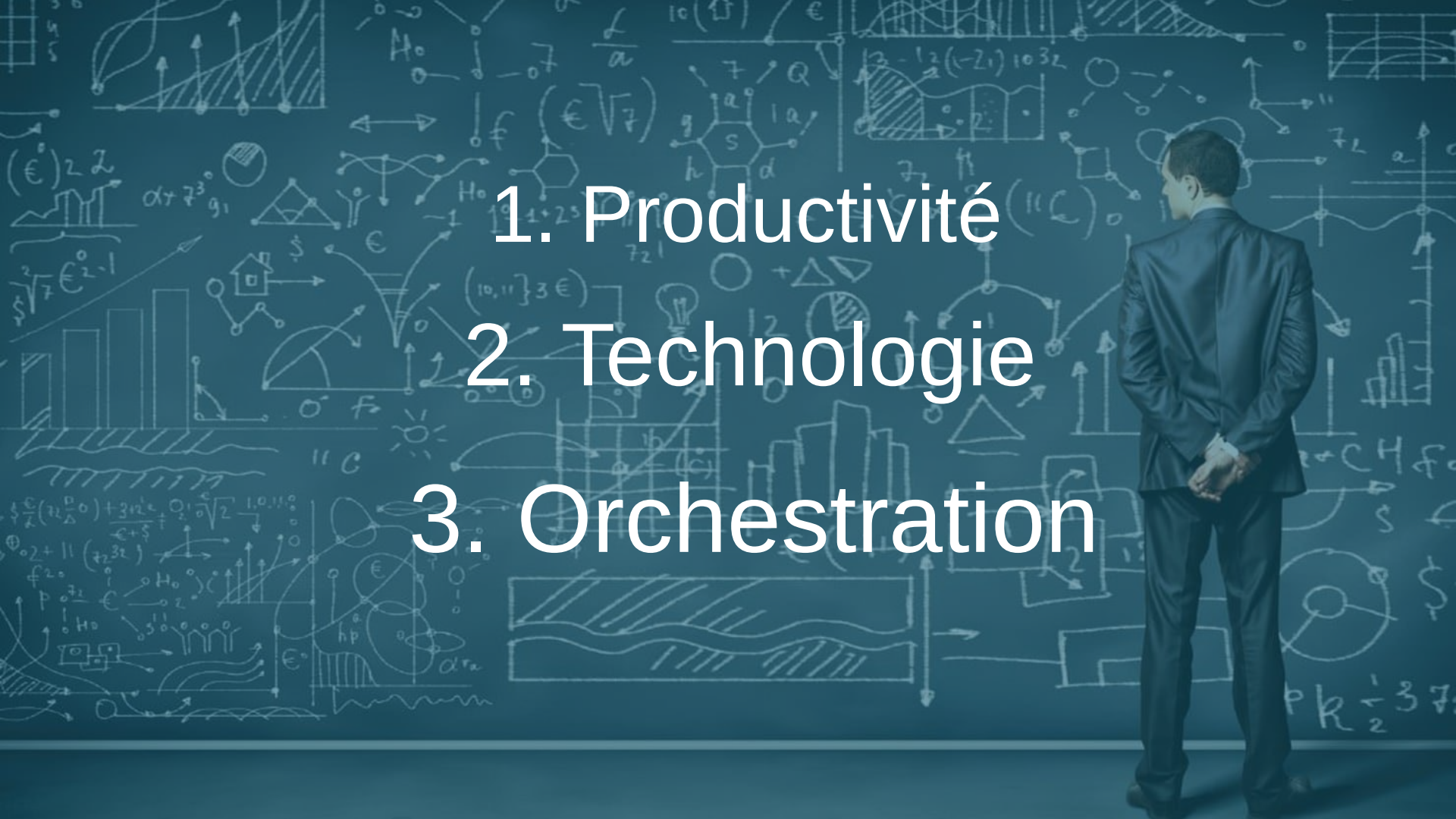
- R
- Python
- Scala
- Java
- Weka
- H20

© Hitachi Vantara Corporation 2018. All Rights Reserved

Pentaho Data Intégration

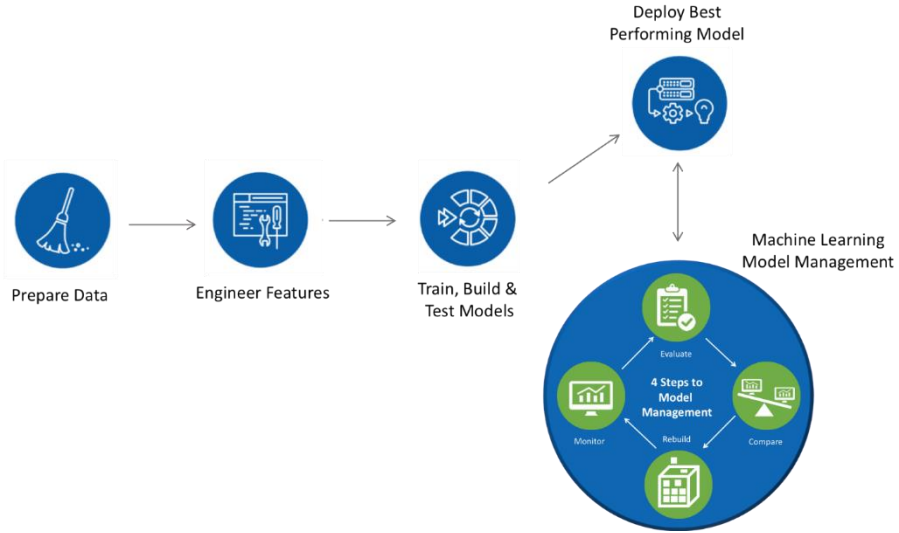


© Hitachi Vantara Corporation 2018. All Rights Reserved


1. Productivité
2. Technologie
3. Orchestration

Pentaho Machine Intelligence
Orchestration et Gestion des Modèles

HITACHI
Inspire the Next



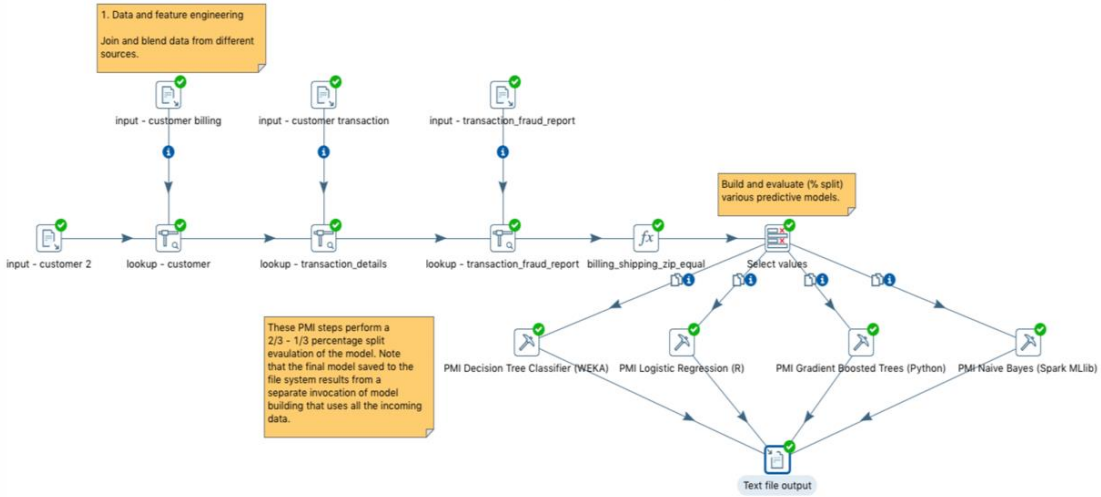
- Accès au machine-learning pour les non-spécialistes
- Plus de productivité pour les data scientistes
  - Préparation de la donnée sans code
  - Algorithmes pré-packagés
- Choix du langage utilisé (Python, R, Spark MLLIB, Weka)
- Orchestration du machine learning en **production :** Capacité à **Monitorer, Evaluer, Comparer et Mettre à jour** les modèles, afin d'accélérer leur déploiement et d'obtenir de meilleurs résultats

© Hitachi Vantara Corporation 2018. All Rights Reserved

Générer les modèles

HITACHI
Inspire the Next

- Data Mining
  - Arff Output
  - Knowledge Flow
  - PMI Decision Tree Classifier
  - PMI Decision Tree Regressor
  - PMI Flow Executor
  - PMI Forecasting
  - PMI Gradient Boosted Trees
  - PMI Linear Regression
  - PMI Logistic Regression
  - PMI Naive Bayes
  - PMI Naive Bayes Multinomial
  - PMI Naive Bayes incremental
  - PMI Random Forest Classifier
  - PMI Random Forest Regressor
  - PMI Scoring
  - PMI Support Vector Classifier
  - PMI Support Vector Regressor
  - Weka Forecasting
  - Weka Scoring



Execution Results

Execution History | Logging | Step Metrics | Performance Graph | Metrics | Preview data

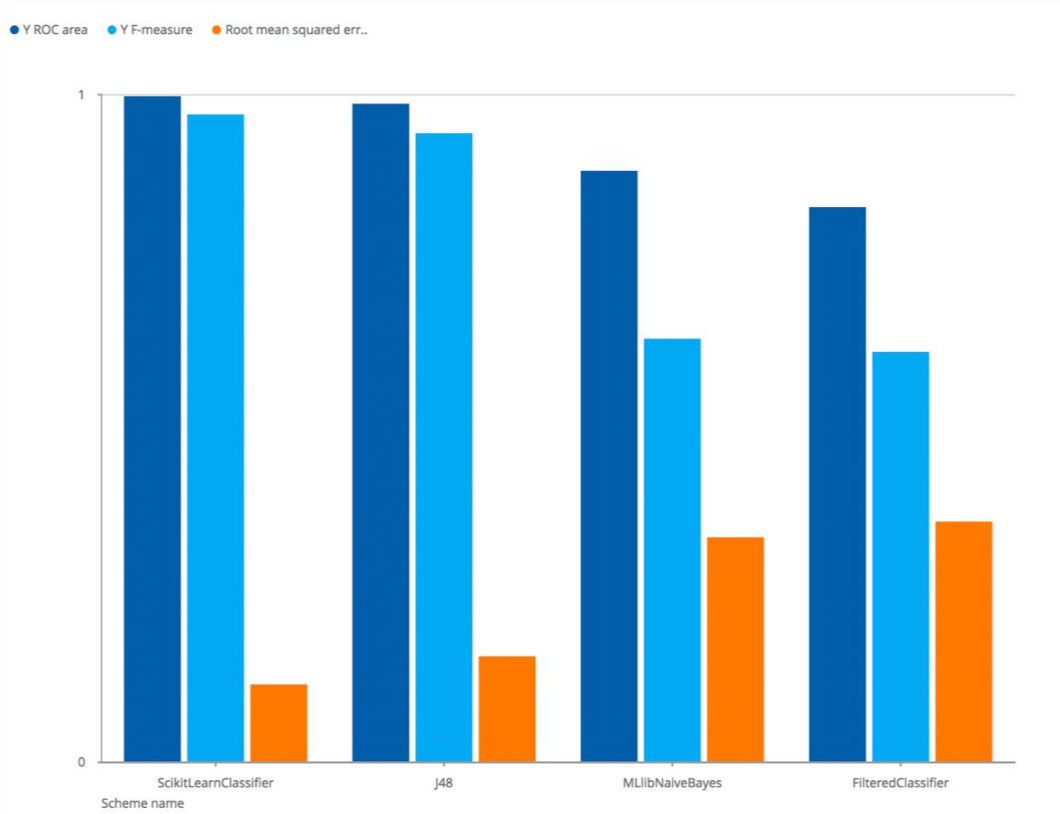
First rows | Last rows | Off

Inspect Data

Scheme name	Scheme options	Evaluation mode	Unclassifie	Correctly classified in	Incorrectly classifie	Percent cor	Percent Inco	Mean absolute erro	Root mean squ	Relative absolu	Root relative squa	Total numb	Kappa stat	N_TP rate	N_FP rate	N_Precision	N_Recall	N_F-measure	N_MCC	Y_
ScikitLearnClassifier	-batch 100 -learner GradientBo...	percentage_split 66% se...	0	33468	532	98.43529	1.56471	0.042	0.11583	10.91204	26.44463	34000	0.95908	0.99123	0.03534	0.9877	0.99123	0.98946	0.95911	0.9
J48	-C 0.25 -M 2	percentage_split 66% se...	0	32997	1003	97.05	2.95	0.04529	0.15826	11.7673	36.13158	34000	0.92258	0.98385	0.06772	0.97652	0.98385	0.98017	0.92267	0.9
MLlibNaiveBayes	-model-type bernoulli -spark-op...	percentage_split 66% se...	0	28616	5384	84.16471	15.83529	0.24368	0.33727	63.31448	77.00022	34000	0.53785	0.95059	0.47029	0.85267	0.95059	0.89897	0.55526	0.5
FilteredClassifier	-F "weka.filters.MultiFilter -F \"w...	percentage_split 66% se...	0	28147	5853	82.78529	17.21471	0.26511	0.3596	68.88182	82.09865	34000	0.50804	0.93111	0.46779	0.85073	0.93111	0.8891	0.51853	0.5

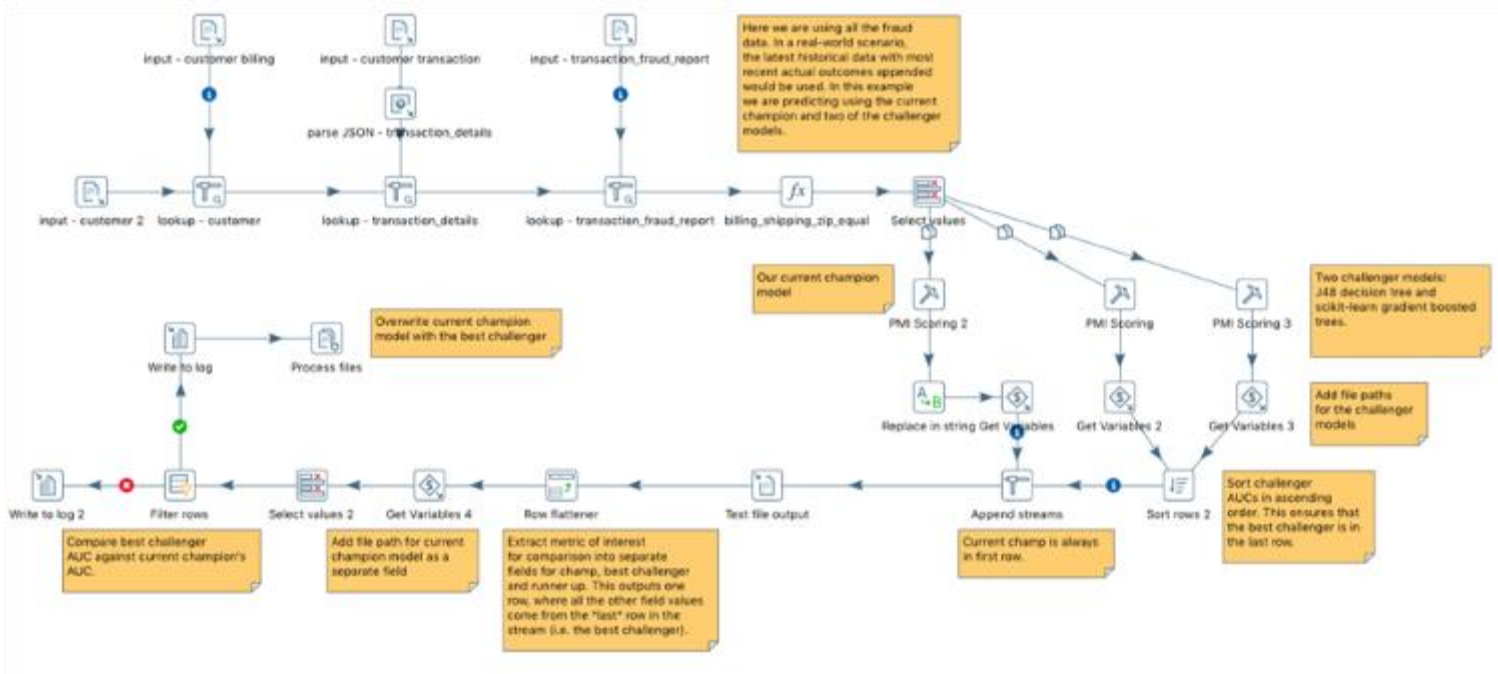
© Hitachi Vantara Corporation 2018. All Rights Reserved

Monitorer la performance des modèles



© Hitachi Vantara Corporation 2018. All Rights Reserved

Challenger les modèles



© Hitachi Vantara Corporation 2018. All Rights Reserved

Roadmap

HITACHI
Inspire the Next

Disponible

No Code

- Decision Tree Classifier – Weka, Python, Spark & R
- Decision Tree Regressor – Weka, Python, Spark & R
- Gradient Boosted Trees – Weka, Python, Spark & R
- Linear Regression – Weka, Python, Spark & R
- Logistic Regression – Weka, Python, Spark & R
- Naive Bayes – Weka, Python, Spark & R
- Naive Bayes Multinomial – Weka, Python & Spark
- Random Forest Classifier – Weka, Python, Spark & R
- Random Forest Regressor – Weka, Python & Spark
- Support Vector Classifier – Weka, Python, Spark & R
- Support Vector Regressor – Weka, Python, & R
- Naive Bayes Incremental – Weka

Coding

- R
- Python
- Spark MLLIB
- Scala / Java

Drag and Drop data-science

- Weka

Bientôt

No Code

- Algorithmes non supervisés
- Deep learning
- Réseaux de neurones

© Hitachi Vantara Corporation 2018. All Rights Reserved

HITACHI
Inspire the Next

Roadmap 8.1

© Hitachi Vantara Corporation 2018. All Rights Reserved

Nouvelles fonctionnalités de la 8.1

HITACHI
Inspire the Next

Cloud and Ecosystem Connectivity

- Connector Updates: Spark, Mongo, Cassandra, Splunk and SFDC
- Google Cloud Platform: Cloud Storage, Big Query and Google Drive
- Amazon Web Services: Enhanced S3, AEL for EMR
- AEL for MapR

Big Data Processing

- ORC File Input and Output
- Improve Avro and Parquet
- Enhanced worker nodes
- Stream Ingestion: MQTT, JMS and WebSphere MQ

Analytics and Insights

- Streaming Data Visualization
- Enhanced Time-series plots
- Enhanced DET Filters: selections

Governance and Administration

- Improved Lineage Export
- Improved Repository Access
- Improved Logging

© Hitachi Vantara Corporation 2018. All Rights Reserved

HITACHI
Inspire the Next

Des questions ?

© Hitachi Vantara Corporation 2018. All Rights Reserved

HITACHI
Inspire the Next

Merci

Jean-François Monteil
Jean.monteil@hitachivantara.com
06 28 66 28 11

© Hitachi Vantara Corporation 2018. All Rights Reserved

HITACHI

Inspire the Next

- Annexes -

HITACHI
Inspire the Next

Cas d’usage

© Hitachi Vantara Corporation 2018. All Rights Reserved

Des métiers transformés par l'IoT

HITACHI
Inspire the Next

	Business Requirements	IoT Analytics	Business Outcomes
CATERPILLAR Marine Asset Management	Analyse télématique	Analyse de la consommation énergétique	Réduction des coûts
Hitachi Rail Europe	Vendre un service	Maintenance prédictive	Gains de plusieurs millions sur la maintenance
PENSKE	Sortir de la maintenance préventive	Maintenance prédictive	Optimisation du taux d'utilisation de la flotte

© Hitachi Vantara Corporation 2018. All Rights Reserved

Monitorer, Evaluer et Comparer les modèles

HITACHI
Inspire the Next

Challenge #1
La precision d'un modèle se degrade avec le temps. Mettre à jour ou changer de modèle sont des operations lourdes.

Maximiser la précision des modèles en production

- Suivre la précision des modèles
- Visualisation de la performance des modèles
- Challenger les nouveaux modèles versus ceux en production

Fonctionalités

- ✓ Métriques d'évaluation pour les algorithmes de régressions et de classification.
  - ✓ Root Mean squared error, AUROC
  - ✓ Coefficient de correlation, Mean Absolute Error

- ✓ L'explorateur de modèles affiche les métriques
- ✓ Un fichier synthétisant les scores est généré

- ✓ Fonctionnalités inclues dans le data pipeline.

© Hitachi Vantara Corporation 2018. All Rights Reserved

(Re)génération des modèles

HITACHI
Inspire the Next

Challenge #2
Il n'existe pas de solution end to end pour mettre en production facilement le ML

Mise en production rapide des modèles	• Paramétrage sans code des modèles • Personnaliser les dernières étapes de la data-prep • Prevention de la surperformance des modèles.

Fonctionnalités

- ✓ 15 algorithmes pré-packagés
- ✓ Interface de paramétrage propre à chaque algorithme
- ✓ R, Python, Weka, MLlib

- ✓ Personnalisation de la préparation des données
  - ✓ Sampling des data
  - ✓ Choix des « features »
  - ✓ Vectorisation des textes

- ✓ Nombreux Modes d'évaluations –
  - ✓ Holdout Strategy
  - ✓ K-fold Cross Validation Strategy
  - ✓ Prequential Strategy

© Hitachi Vantara Corporation 2018. All Rights Reserved

Deploy les modèles

HITACHI
Inspire the Next

Challenge #3
Manque de collaboration, de scalabilité et de gouvernance

Collaborer, Scaler et appliquer de la gouvernance à la mise en production des Modèles	• Plateforme collaborative proposant des fonctionnalité pour tous les acteurs de la chaine de données. • Plateforme dédiée aux entreprises

Fonctionnalités

- ✓ Data lineage (data provenance)
  - ✓ Data Sources
  - ✓ Model Features
  - ✓ Parameters & Coefficients

- ✓ Plateforme d'intégration
  - ✓ Scalable
  - ✓ Distribuée
  - ✓ Respectueuse de Sécurité et des ACL des données

© Hitachi Vantara Corporation 2018. All Rights Reserved