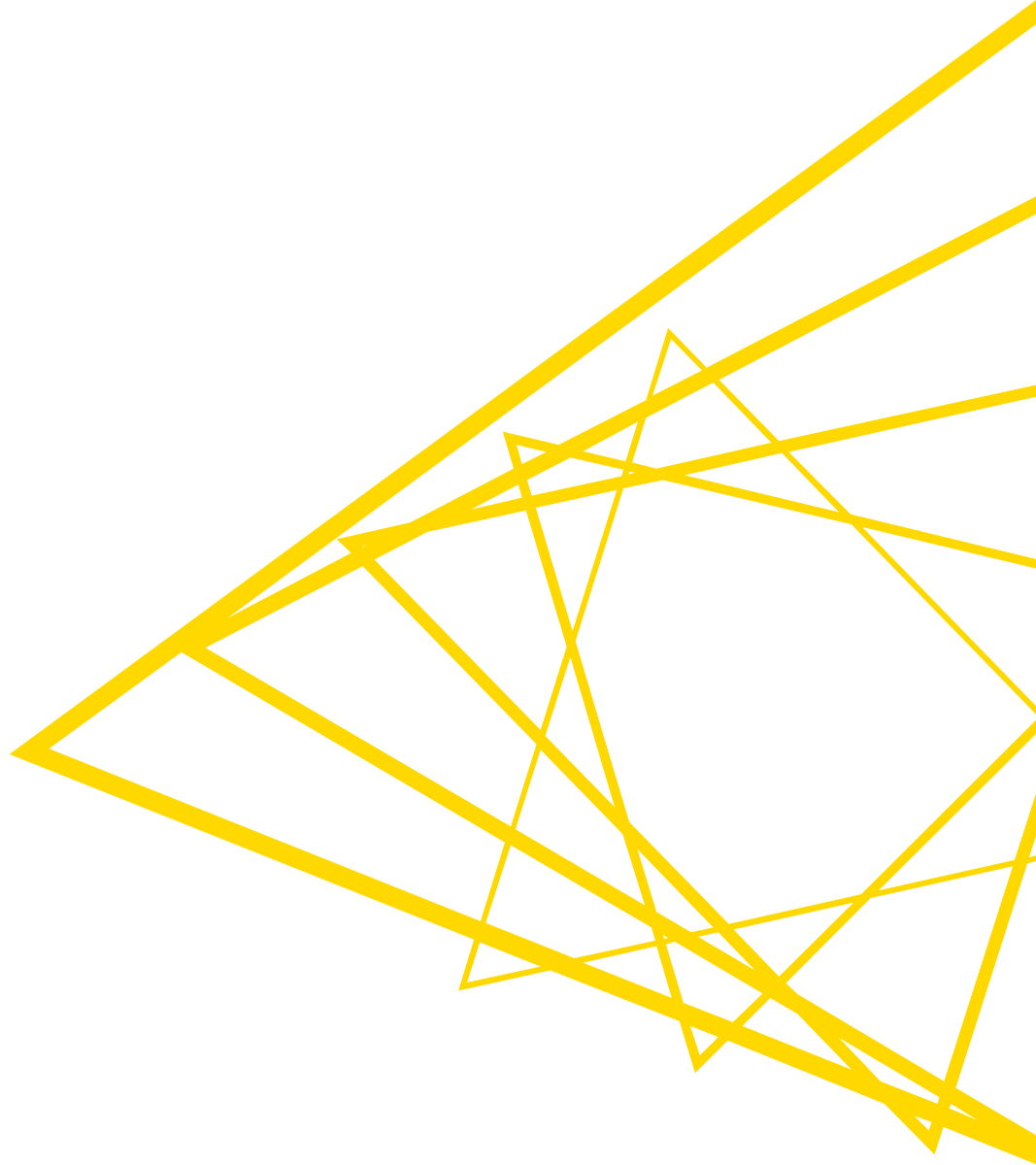




# On monsters and tags...

Jeanette Prinz  
Greg Landrum  
KNIME



# Fun with Tags

Blog post: <https://www.knime.com/blog/fun-with-tags>

Workflow:

knime://EXAMPLES/08\_Other\_Analytics\_Types/02\_Chemistry\_and\_Life\_Sciences/03\_Fun\_with\_Tags

# We can't rely on other people's graffiti

---

Banksy Painting Self-Destructs After Fetching \$1.4 Million at Sotheby's



<https://www.nytimes.com/2018/10/06/arts/design/uk-banksy-painting-sothebys.html>

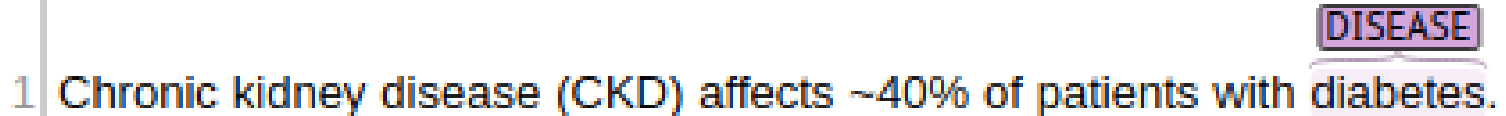
<https://consequenceofsound.net/2018/10/banksy-painting-self-destructs/>

# Text Tagging

---

Text tagging is the process of adding a tag (language element descriptor) to unstructured data

1 Chronic kidney disease (CKD) affects ~40% of patients with diabetes.

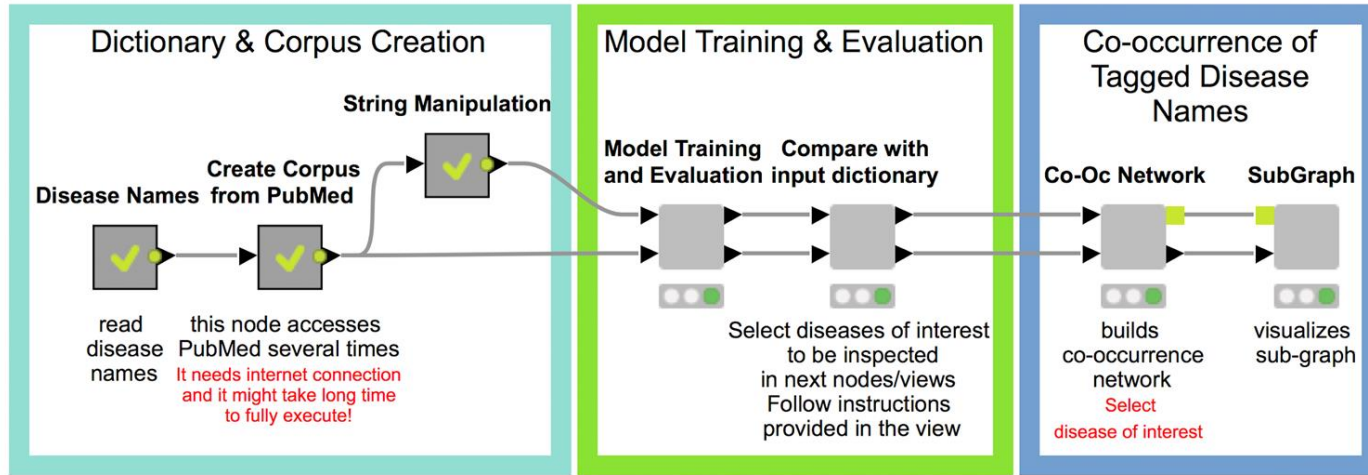


# Motivation

---

- Automated access to disease information is an important goal of text-mining efforts
- Rapid growth of biomedical literature => impossible for humans alone to extract and exhaust all of the useful information
- Enables integration with other data types and the generation of new hypotheses

# The Workflow



Dictionary: contains disease names from [Ensembl Biomart](#) assembled from different sources such as [OMIM](#), [Orphanet](#), and [DDG2P](#).

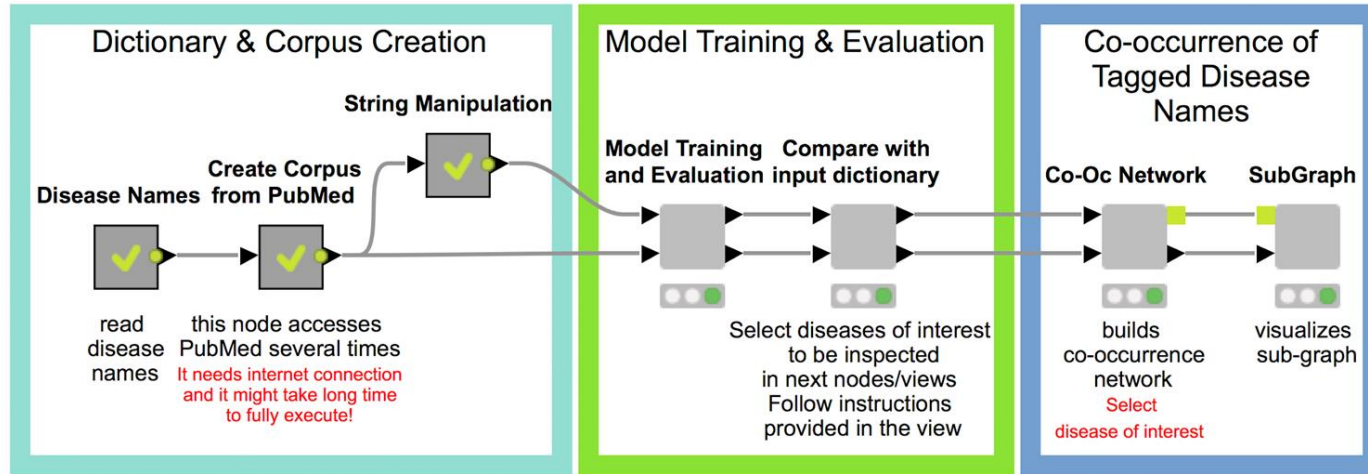
# Corpus Creation?

---



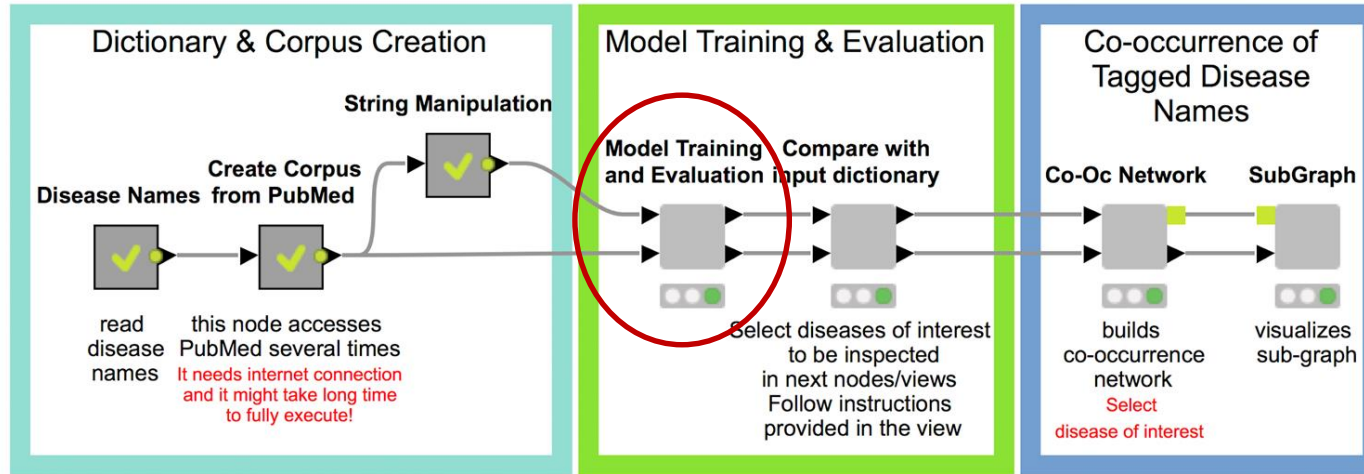
- In linguistics, a **corpus** is a large and structured set of texts (usually electronically stored and processed) ([https://en.wikipedia.org/wiki/Text\\_corpus](https://en.wikipedia.org/wiki/Text_corpus))
- KNIME node “Document Grabber” allows to extract abstracts from [PubMed](#) containing disease names from our dictionary. We only keep results with min. 20 hits in PubMed, and collect max. 100 documents per disease

# The Workflow





# The Workflow



---



# Demo

# Ohdo Syndrome - KAT6B

AJHG

REPORT | [VOLUME 89, ISSUE 5, P675-681, NOVEMBER 11, 2011](#)

## Whole-Exome-Sequencing Identifies Mutations in Histone Acetyltransferase Gene *KAT6B* in Individuals with the Say-Barber-Biesecker Variant of Ohdo Syndrome

[Jill Clayton-Smith](#)   • [James O'Sullivan](#) • [Sarah Daly](#) • ... [Shehla Mohammed](#) • [Dian Donnai](#) • [Graeme Black](#) • [Show all authors](#)

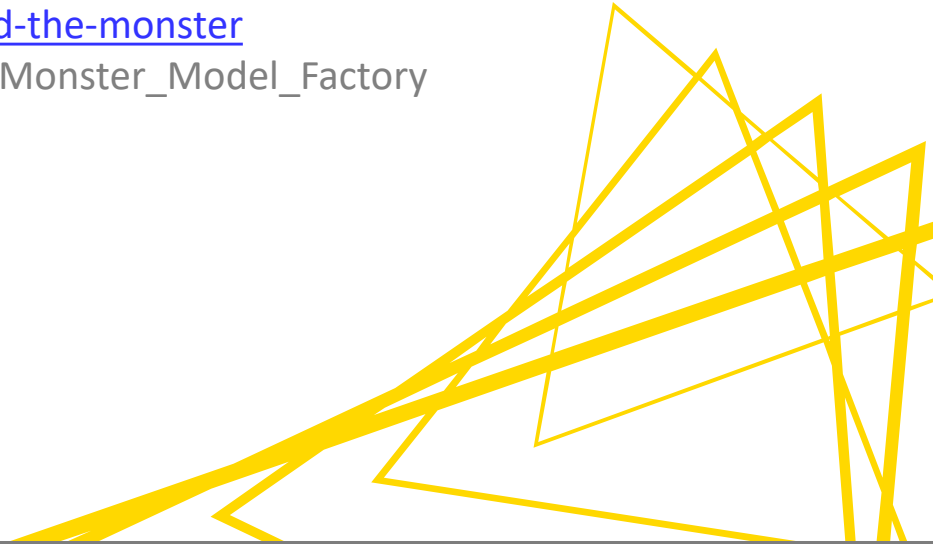
[Open Archive](#) • Published online: November 10, 2011 • DOI: <https://doi.org/10.1016/j.ajhg.2011.10.008>

# Monster Model Factory

Video: <https://bit.ly/2O8Glqs>

Blog post: <https://www.knime.com/blog/beauty-and-the-monster>

Workflow: `knime://EXAMPLES/50_Applications/37_Monster_Model_Factory`



## Build, validate, and deploy models for multiple data sets

---

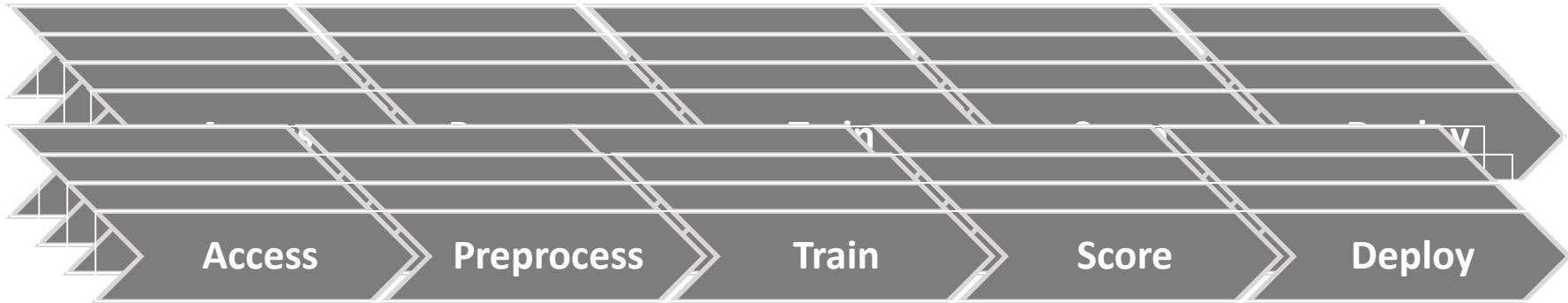
- >1500 datasets from ChEMBL that I would like to build models for
- I want to actually use the models, so they need to be deployed
- The whole process needs to be automated and reproducible so that I can do it again when ChEMBL is updated
- Maybe we can learn something interesting from the models themselves

# The model process

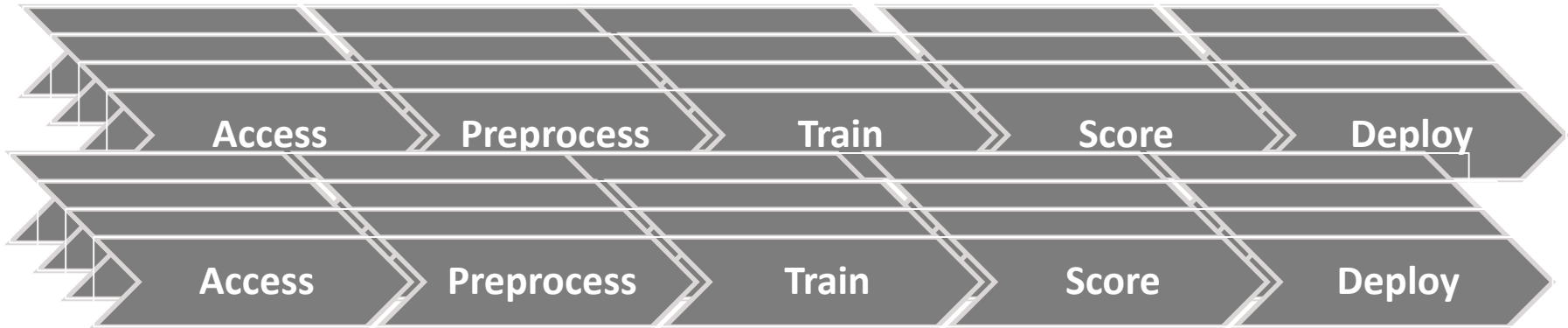
---



# The model process, multiple models

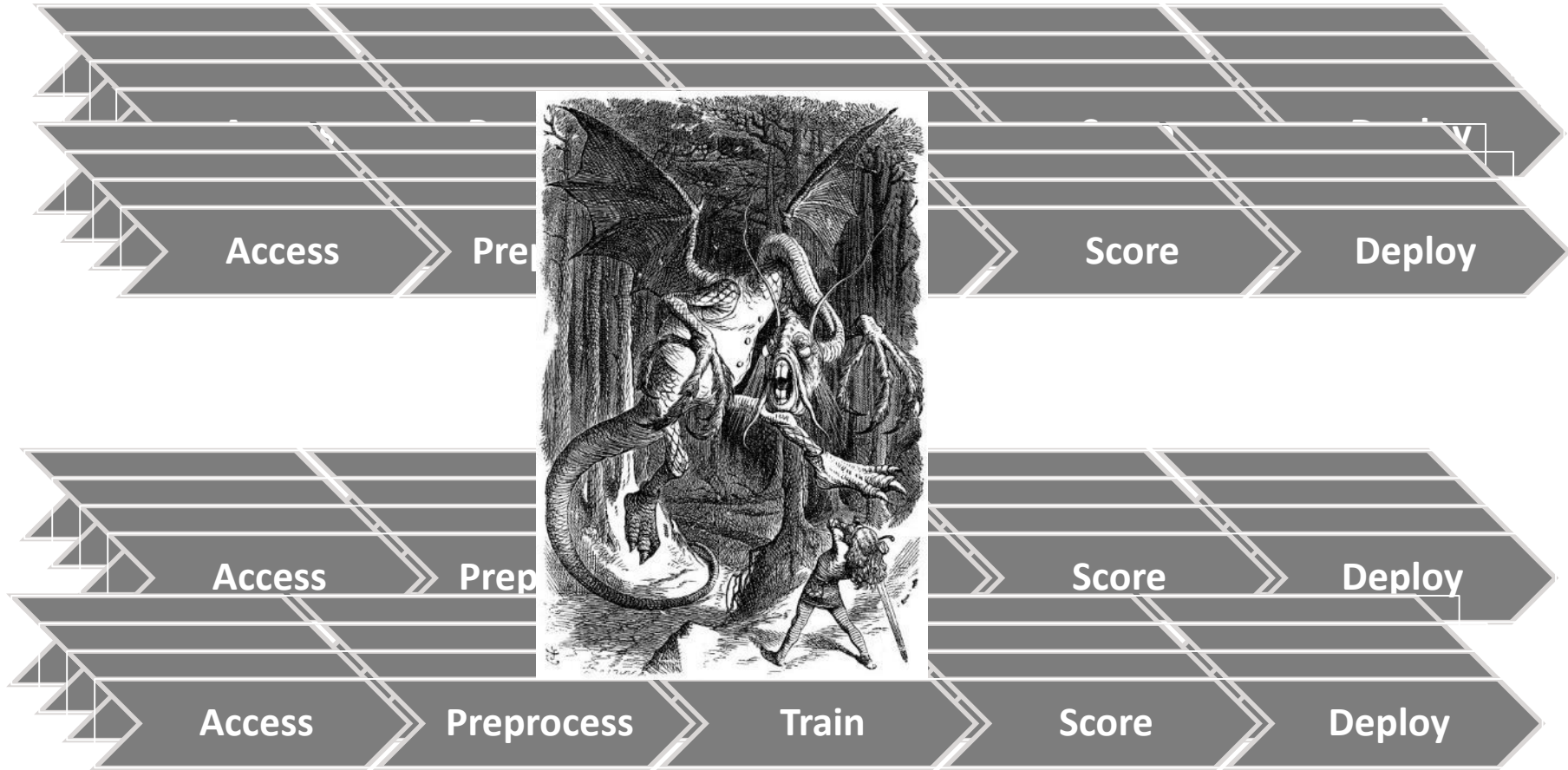


...



<https://commons.wikimedia.org/wiki/File:Jabberwocky.jpg>

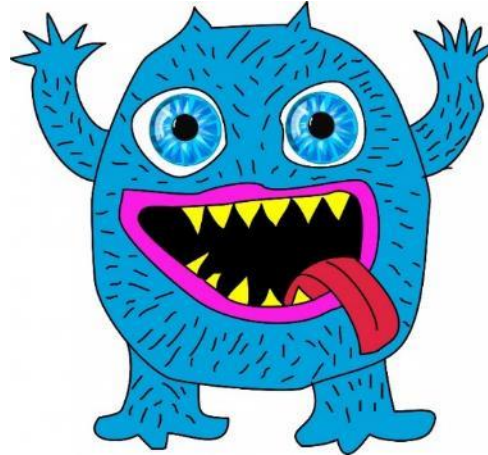
# The model process, multiple models



<https://commons.wikimedia.org/wiki/File:Jabberwocky.jpg>



# AUTOMATE



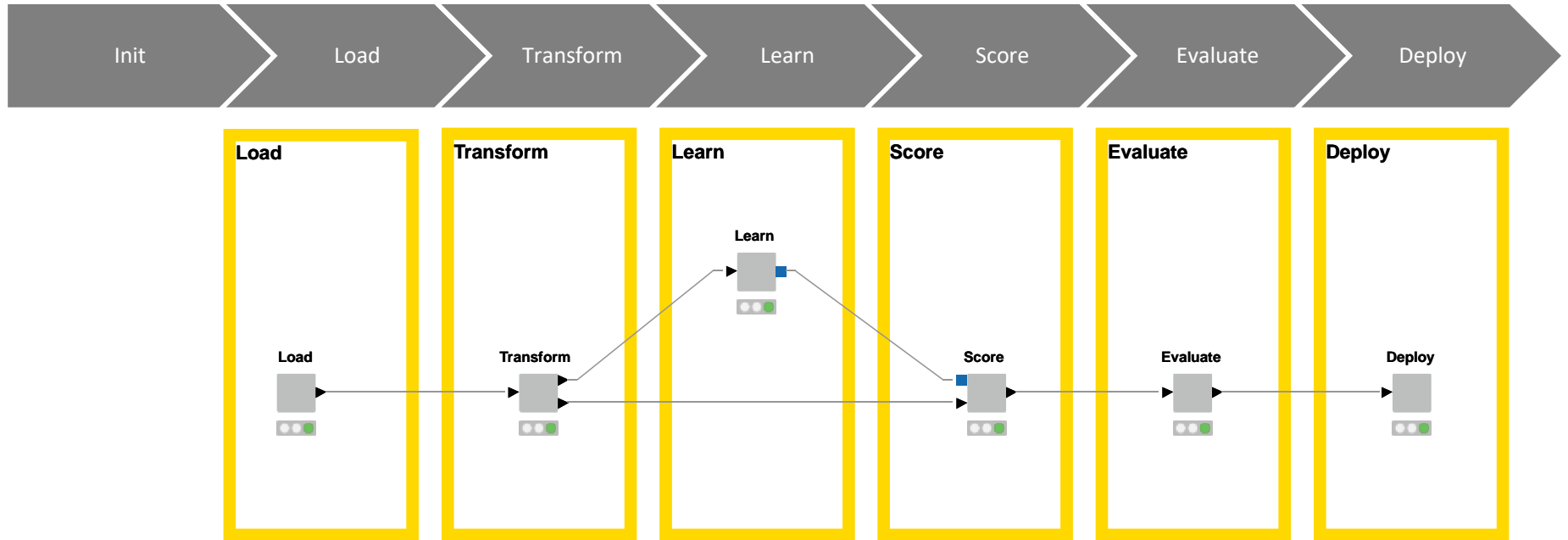
<https://www.publicdomainpictures.net/view-image.php?image=155188>

# ALL THE THINGS!

# Automation: the model process factory

---

# Automation: the model process factory



Make each step a separate workflow.

Use KNIME to orchestrate calling those workflows

KNIME blog post: <https://goo.gl/LvESqB>

White paper: <https://goo.gl/d6UpUu>

# Monster Model Factory

Init

Access

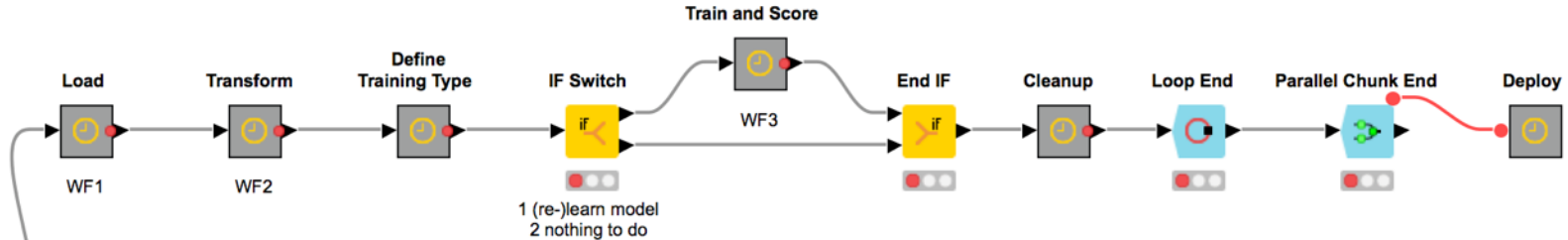
Preprocess

Train

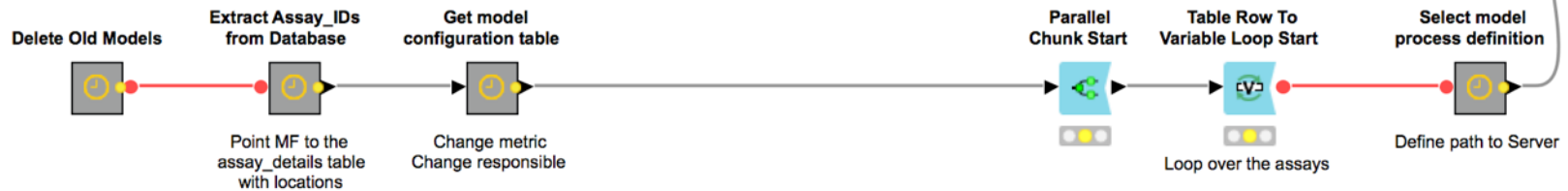
Score

Deploy

## Abstract Model Factory



## Prepare Modelling Configuration

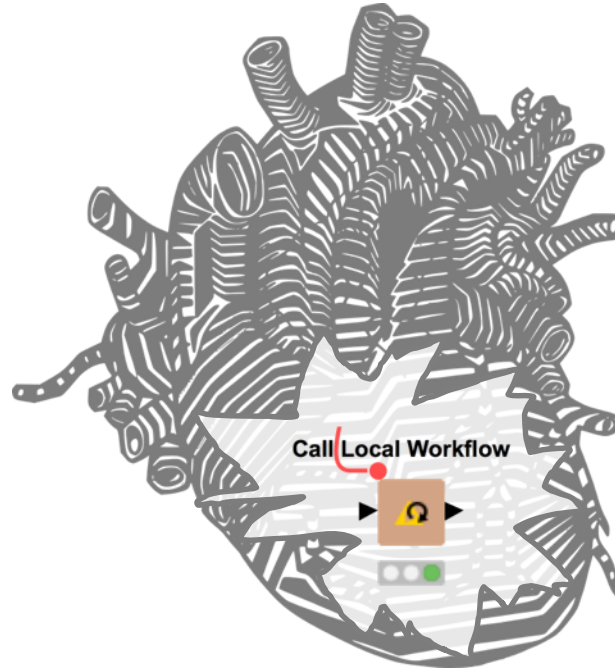


Monster Model Factory workflow:

[knime://EXAMPLES/50 Applications/37 Monster Model Factory](https://knime.com/EXAMPLES/50_Applications/37_Monster_Model_Factory)

# The heart of the factory: Call Local Workflow<sup>1</sup>

---



<sup>1</sup> Call Remote Workflow when run on the KNIME Server<sup>2</sup>

<sup>2</sup>If we were doing this today, we'd use the Call Workflow node.

- Executes another workflow in the same local repository

<https://pixabay.com/en/heart-veins-arteries-anatomy-152594/>

# Monster Model Factory

Init

Access

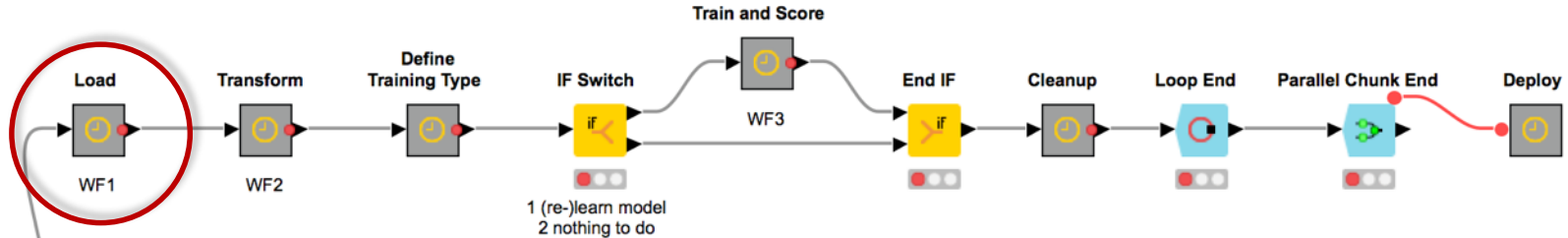
Preprocess

Train

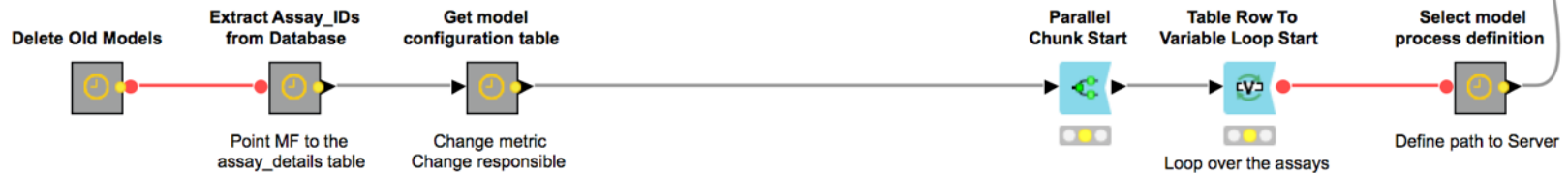
Score

Deploy

## Abstract Model Factory



## Prepare Modelling Configuration



Monster Model Factory workflow:

[knime://EXAMPLES/50 Applications/37 Monster Model Factory](https://knime.com/EXAMPLES/50_Applications/37_Monster_Model_Factory)

# Monster Model Factory

Init

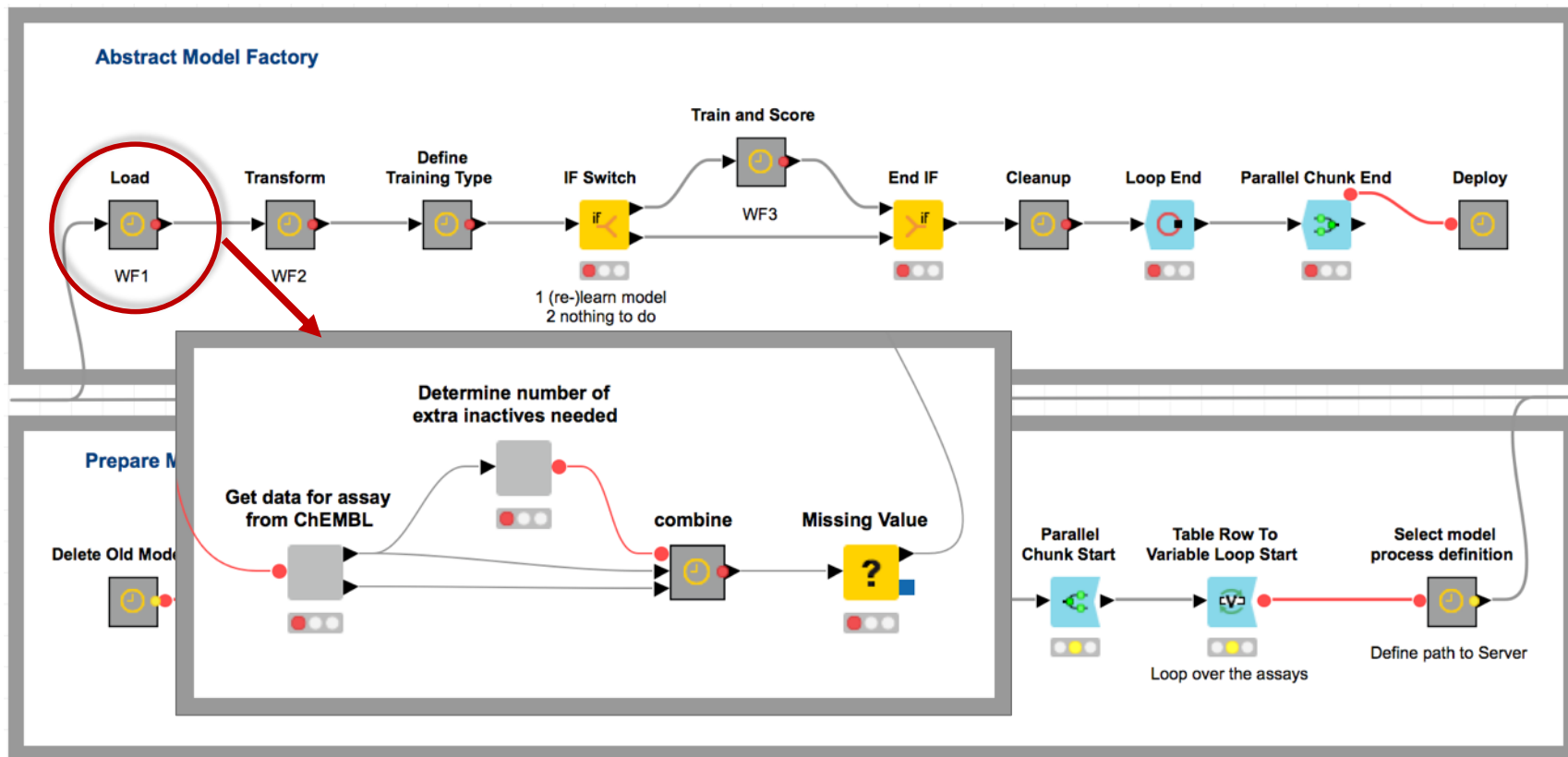
Access

Preprocess

Train

Score

Deploy



Monster Model Factory workflow:

[knime://EXAMPLES/50 Applications/37 Monster Model Factory](https://knime.org/EXAMPLES/50_Applications/37_Monster_Model_Factory)

# Monster Model Factory

Init

Access

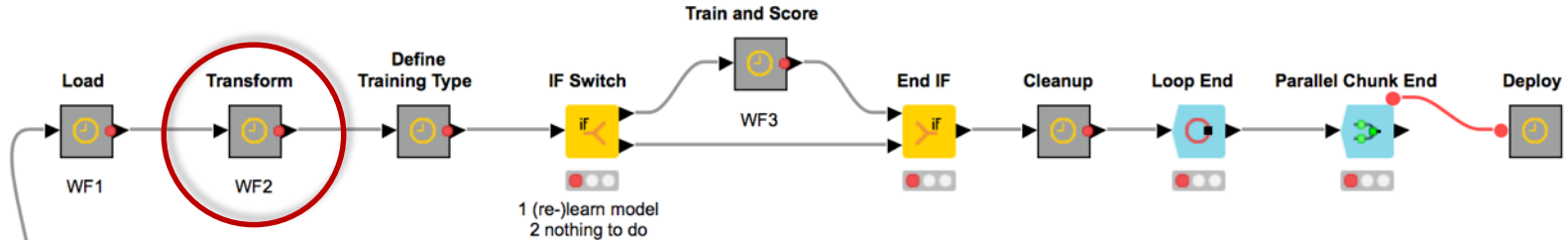
Preprocess

Train

Score

Deploy

## Abstract Model Factory



## Prepare Modelling Configuration



Monster Model Factory workflow:

[knime://EXAMPLES/50 Applications/37 Monster Model Factory](https://knime.com/EXAMPLES/50_Applications/37_Monster_Model_Factory)



# Monster Model Factory

Init

Access

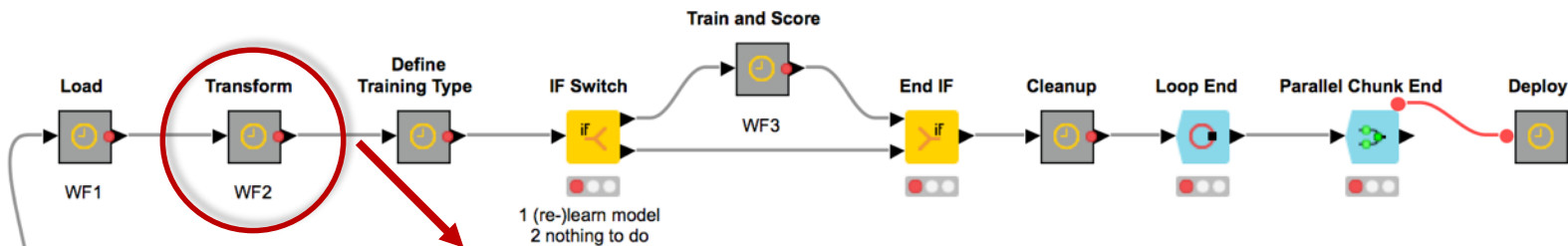
Preprocess

Train

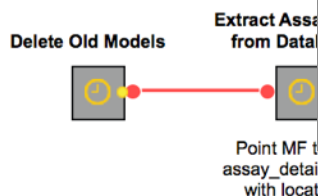
Score

Deploy

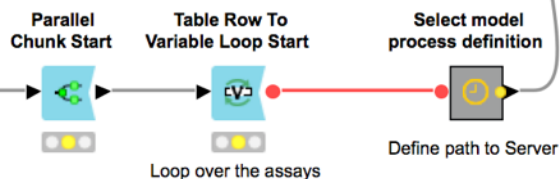
## Abstract Model Factory



## Prepare Modelling Configuration



## Preprocess Structures



Monster Model Factory workflow:

[knime://EXAMPLES/50 Applications/37 Monster Model Factory](https://knime.org/EXAMPLES/50_Applications/37_Monster_Model_Factory)

# Monster Model Factory

Init

Access

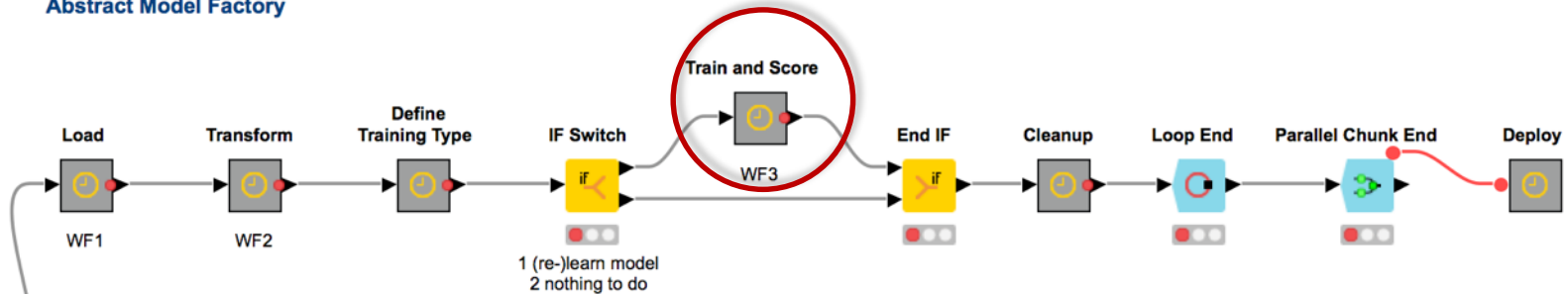
Preprocess

Train

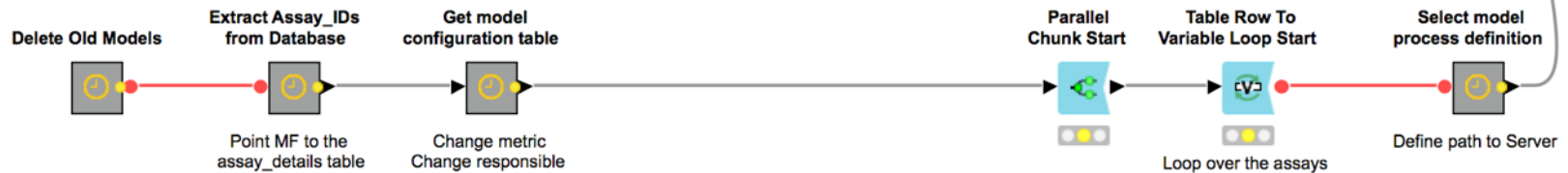
Score

Deploy

## Abstract Model Factory



## Prepare Modelling Configuration



Monster Model Factory workflow:

[knime://EXAMPLES/50 Applications/37 Monster Model Factory](https://knime.com/EXAMPLES/50_Applications/37_Monster_Model_Factory)

# Monster Model Factory

Init

Access

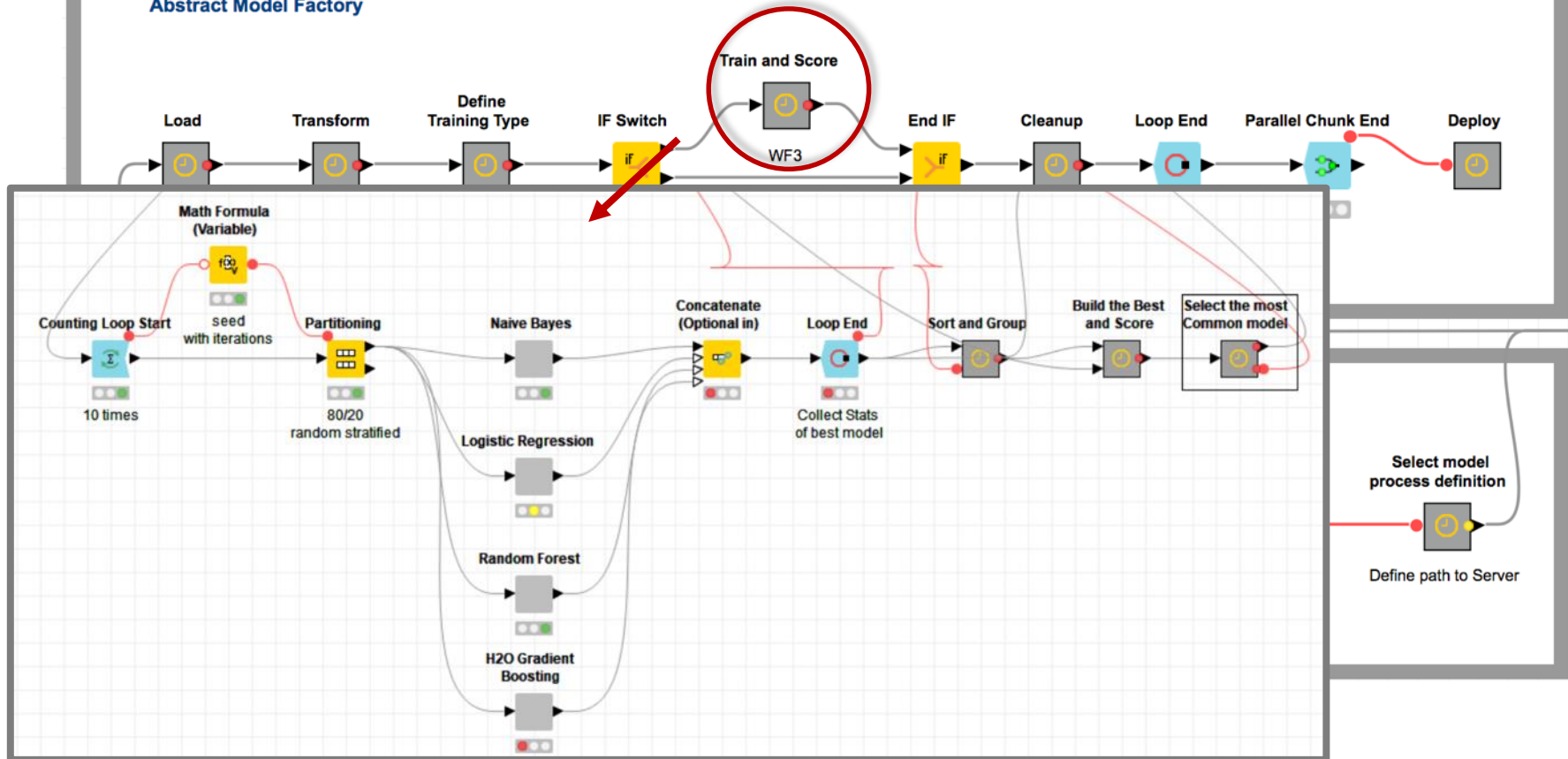
Preprocess

Train

Score

Deploy

## Abstract Model Factory



[KNIME://EXAMPLES/50\\_Applications/57\\_Monster\\_Model\\_Factory](#)

# Monster Model Factory

Init

Access

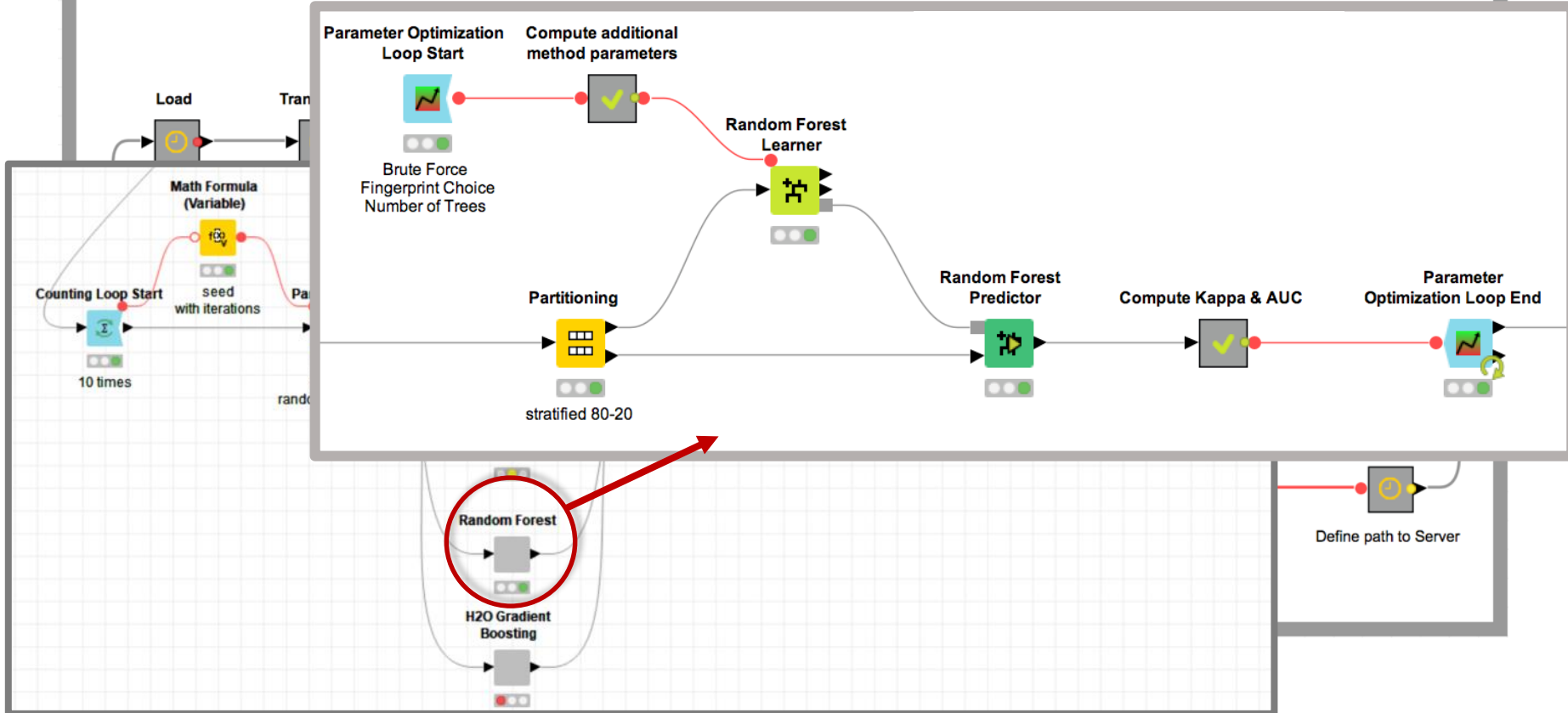
Preprocess

Train

Score

Deploy

## Abstract Model Factory



[knime.org/EXAMPLES/50-Applications/57-Monster-Model-Factory](http://knime.org/EXAMPLES/50-Applications/57-Monster-Model-Factory)

# Monster Model Factory

Init

Access

Preprocess

Train

Score

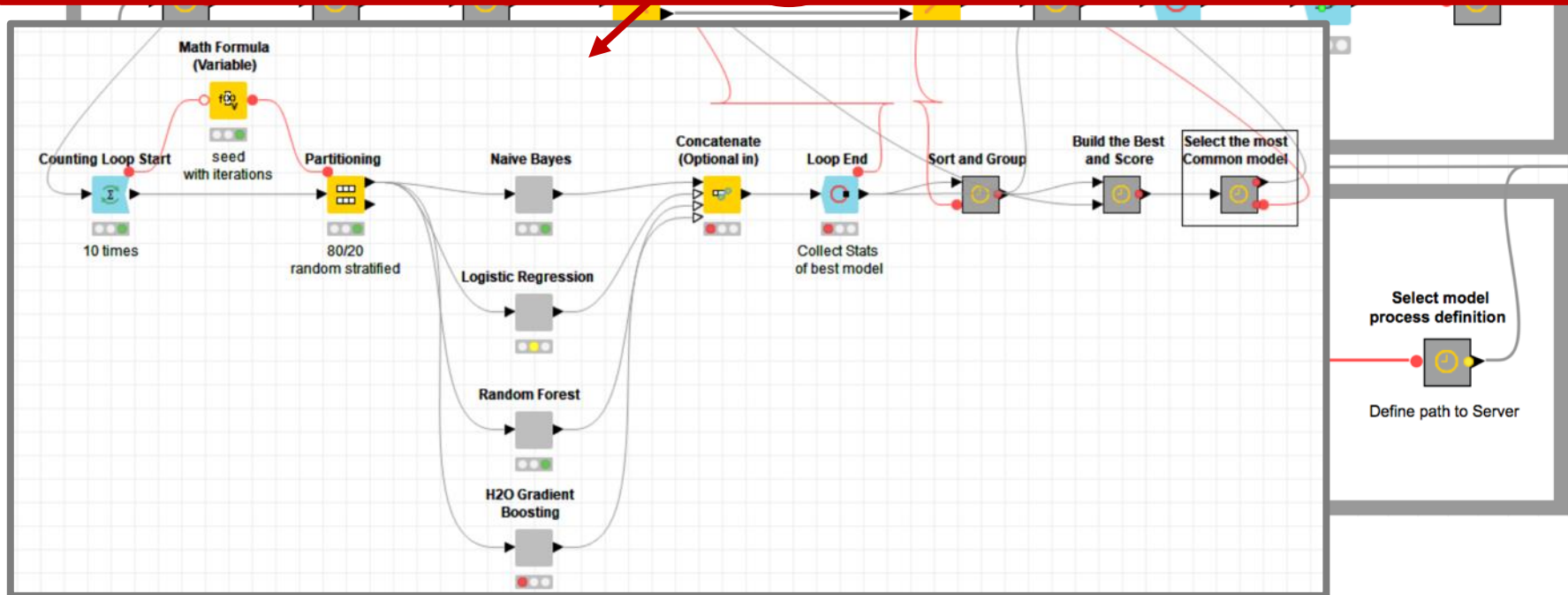
Deploy

Blog post:

<https://www.knime.com/blog/stuck-in-the-nine-circles-of-hell-try-parameter-optimization-a-cup-of-tea>

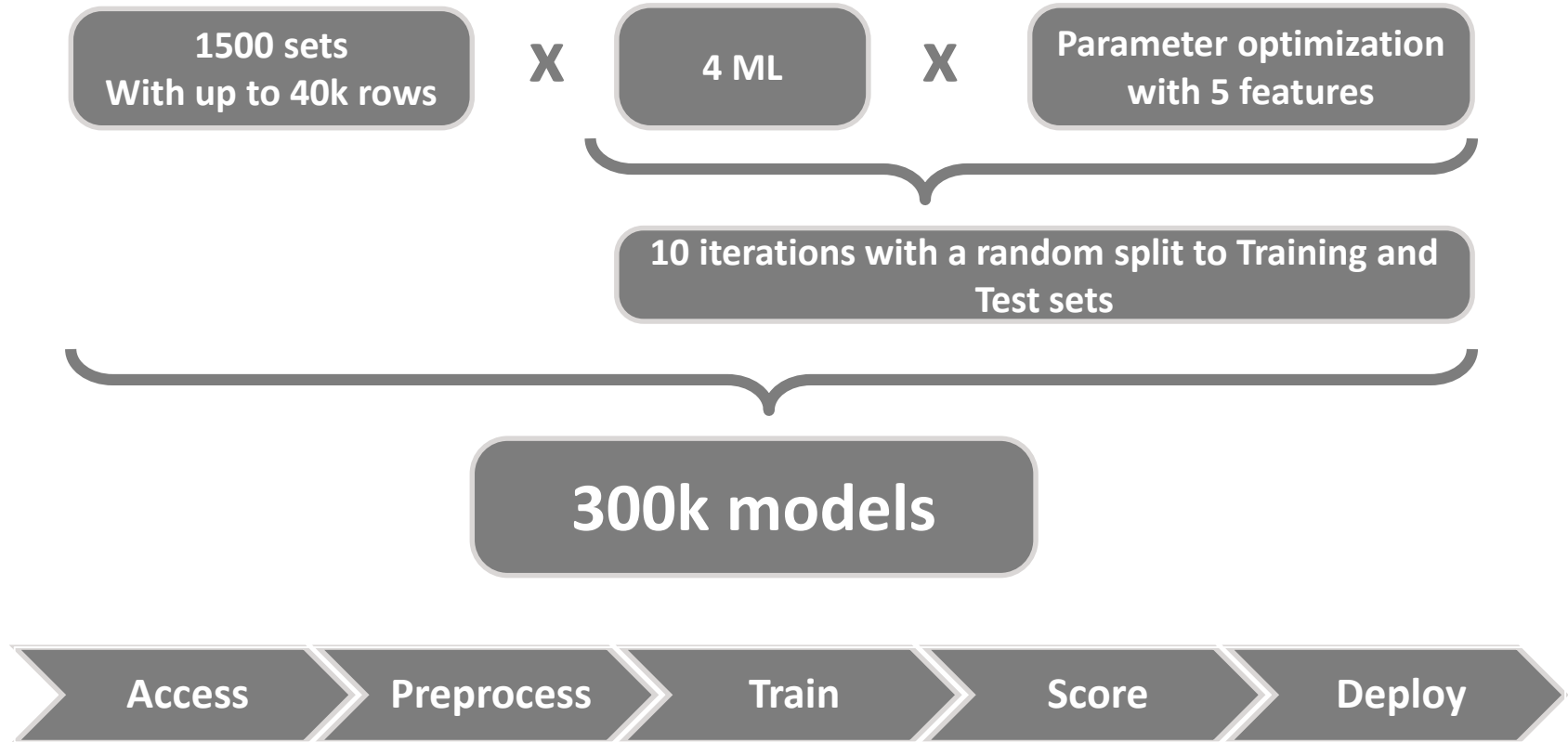
Workflow:

knime://EXAMPLES/04\_Analytics/11\_Optimization/08\_Model\_Optimization\_and\_Selection

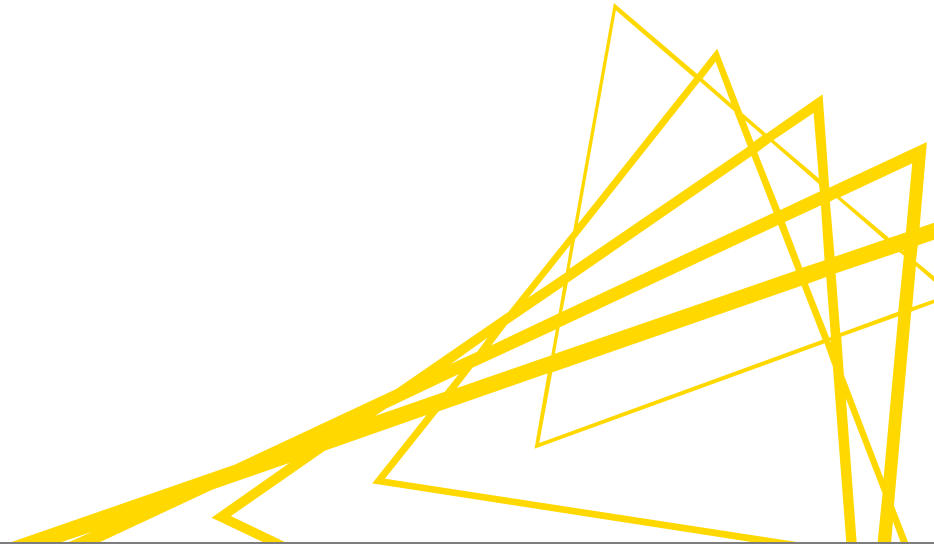


[knime://EXAMPLES/04\\_Analytics/11\\_Optimization/08\\_Model\\_Optimization\\_and\\_Selection](https://www.knime.com/blog/stuck-in-the-nine-circles-of-hell-try-parameter-optimization-a-cup-of-tea)

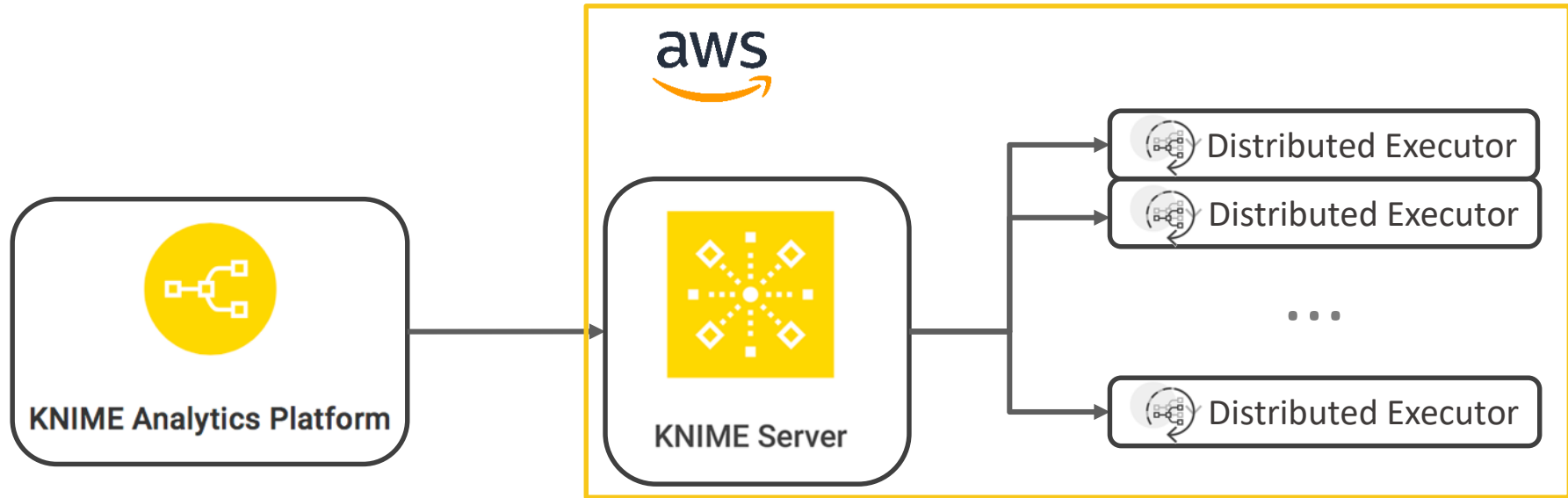
# Task Scale



# Scaling Execution



# Scale with KNIME Server Distributed Executors



Build/test workflows

Run model factory

Run individual data sets  
65-70 load-balanced  
distributed executors



# Model Factory

Init

Access

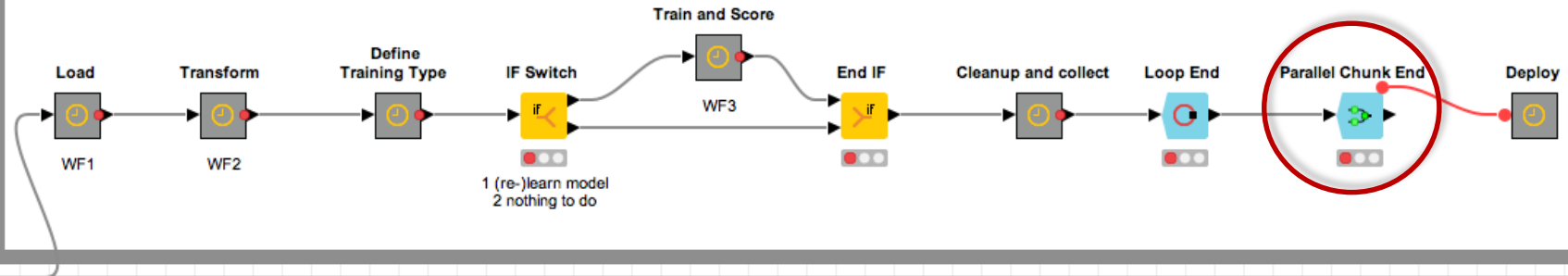
Preprocess

Train

Score

Deploy

## Abstract Model Factory



## Prepare Modelling Configuration



# Summary

---

- Automated the Model Management Process using KNIME Analytics Platform
- In total 300K models were built<sup>1</sup>
- Scaled it for our task with KNIME Server distributed executors
- Less than a week<sup>2</sup>

<sup>1</sup> ~1550 assays \* 4 methods \* 5 FPs \* 10 repeats

<sup>2</sup> ~4400 hours computing time on single CPU

Video: <https://bit.ly/2O8GIqs>

Blog post: <https://www.knime.com/blog/beauty-and-the-monster>

Workflow: `knime://EXAMPLES/50_Applications/37_Monster_Model_Factory`

The KNIME® trademark and logo and OPEN FOR INNOVATION® trademark are used by KNIME AG under license from KNIME GmbH, and are registered in the United States. KNIME® is also registered in Germany.