

# Open standards for machine learning model deployment

## IBM Developer

**Svetlana Levitan, PhD**

Senior Developer Advocate

Center for Open Data and AI Technologies (CODAIT)

IBM Cognitive Applications



# Who is Svetlana Levitan?

Originally from Moscow, Russia

PhD in Applied Mathematics and MS in Computer Science from University of Maryland, College Park

Software Engineer for SPSS Analytic components (2000-2018)

Working on PMML since 2001, ONNX recently

IBM acquired SPSS in 2009

Developer Advocate with IBM Center for Open Data and AI Technologies (since June 2018)

Meetup organizer: Big Data Developers in Chicago, Open Source Analytics, IBM Cloud, Chicago ML

Two daughters love programming

IBM Developer

@**SvetaLevitan**

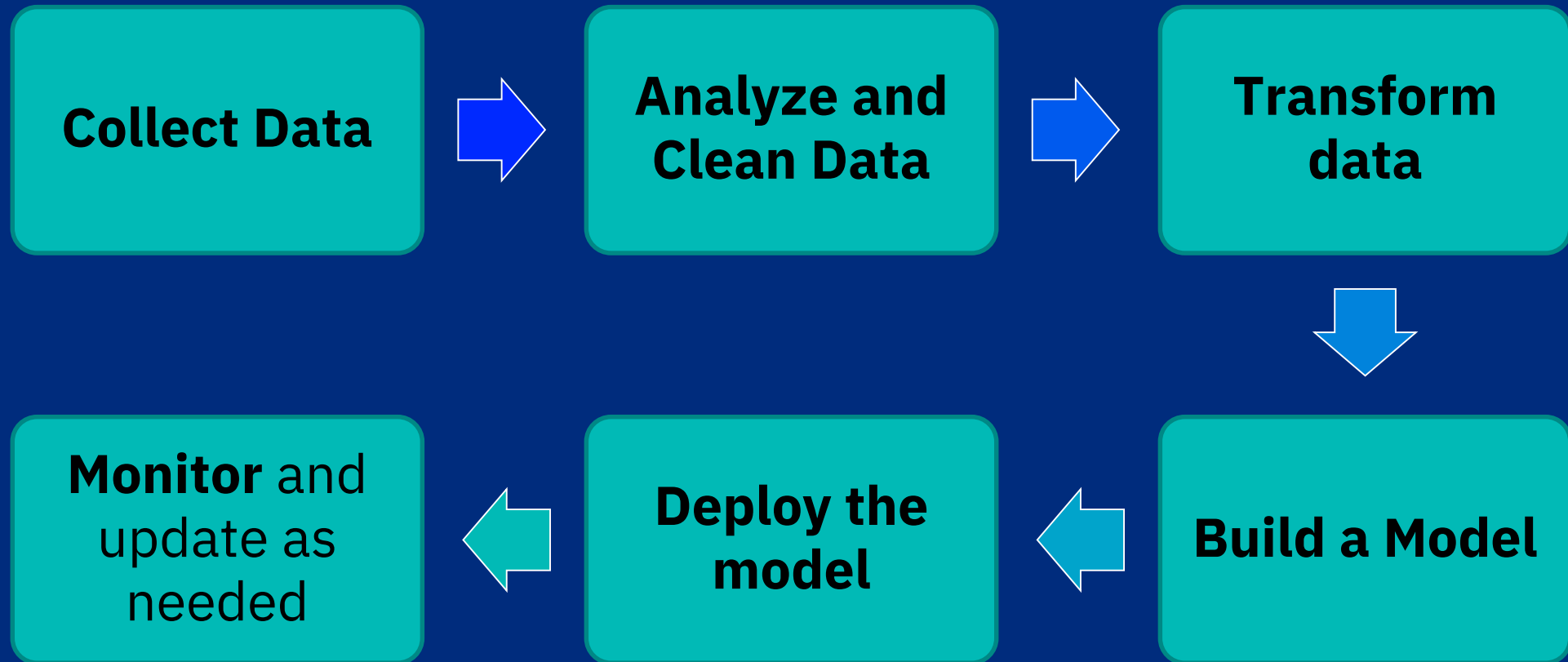




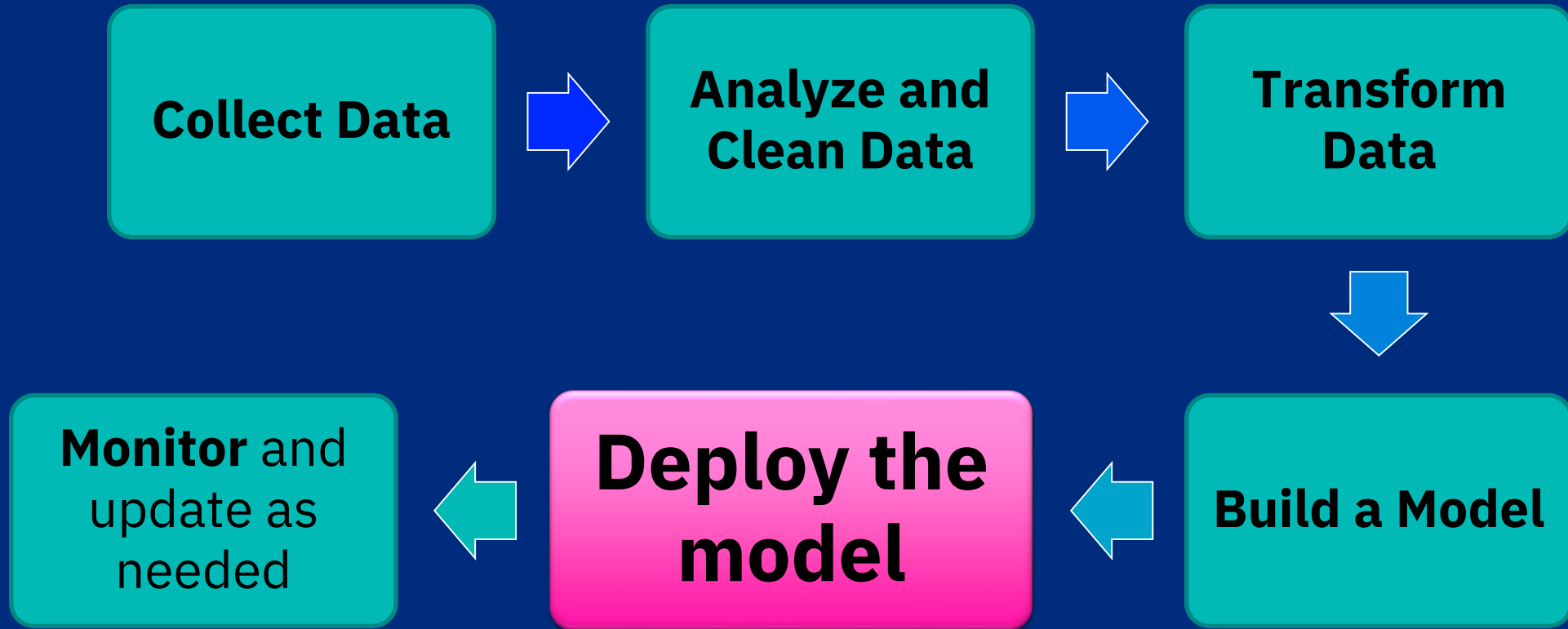
# Agenda

- **Deployment challenges**
- **PMML Internals**
- **PMML in Python and R**
- **PMML in IBM products**
- **PFA**
- **ONNX**

# Typical Stages in Machine Learning



# Typical Stages in Machine Learning



# Model Deployment Challenges

## Teams

- Data Scientists and statisticians
- Application developers and IT

## Environments

- OS and File Systems
- Databases, desktop, cloud

## Languages

- Python or R, various packages, C++ or Java or Scala, Dependencies and versions

## Data Preparation

- Aggregation and joins
- Normalization, Category Encoding, Binning, Missing value replacement



# DMG to the rescue!



Data Mining Group since 1990's

[dmg.org](http://dmg.org)

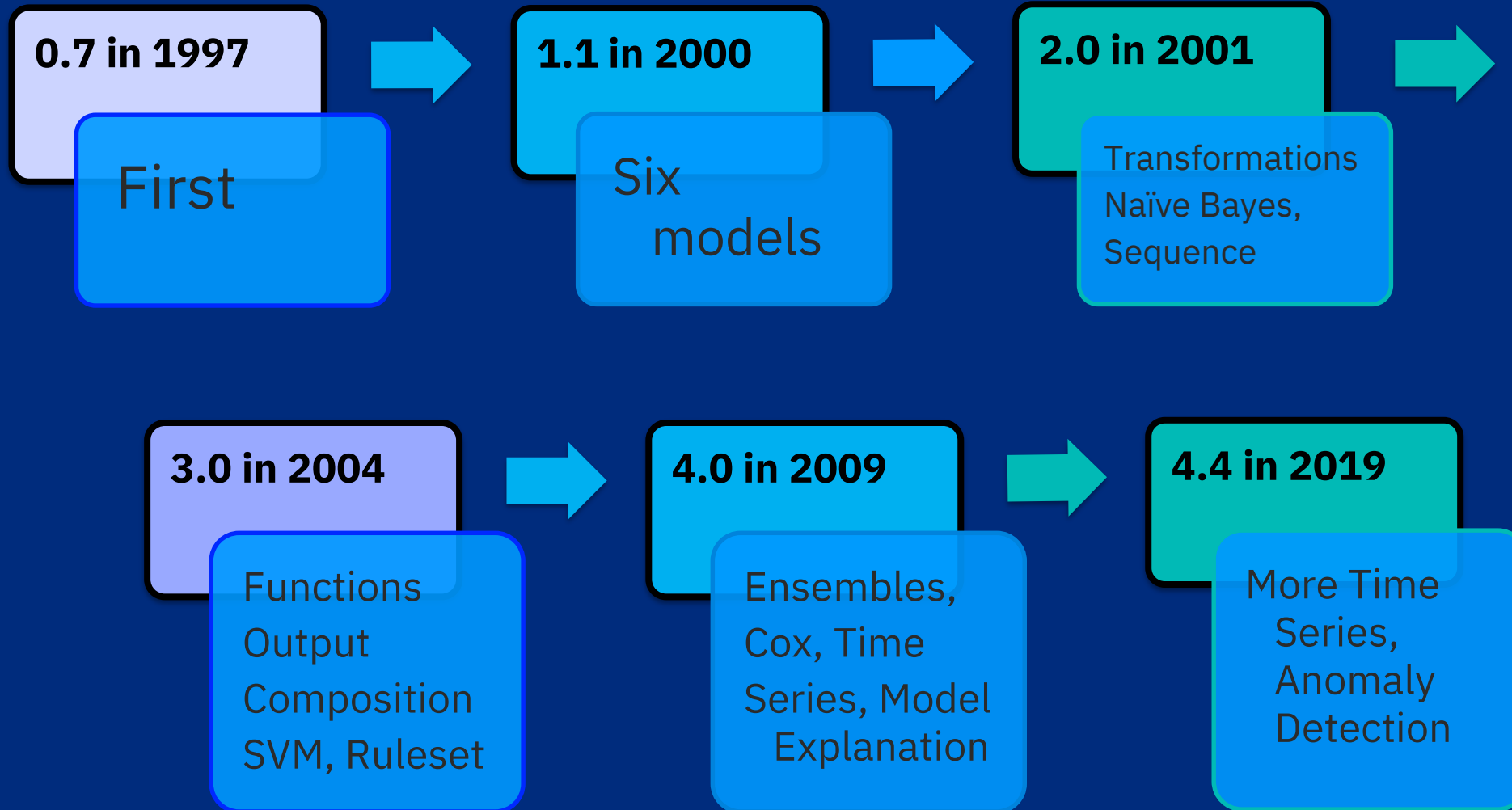


## Predictive Model Markup Language

- An Open Standard for **XML** Representation
- Over 30 vendors and organizations
- PMML 4.4 Release manager: Svetlana Levitan



# Brief History of PMML versions





# Main Components of PMML



Header

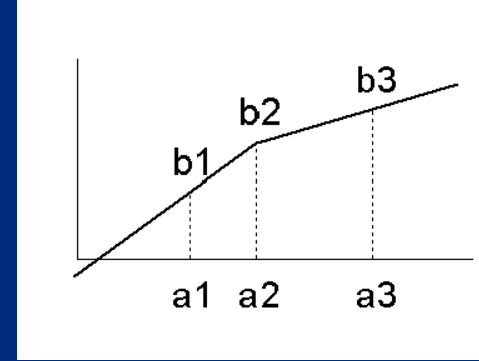
Data Dictionary

Transformation Dictionary

Model(s)

# Transformations

- **NormContinuous:** piece-wise linear transform
- **NormDiscrete:** map a categorical field to a set of dummy fields
- **Discretize:** binning
- **MapValues:** map one or more categorical fields into another categorical one
- **Functions:** built-in and user-defined
- Other transformations



\*favorite\_pets.sav [DataSet2] - IBM SPSS Statistics

pet\_d2 is 1 if (pet = 2), 0 otherwise.

	pet	pet_d1	pet_d2	pet_d3	pet_d4
1	1	1	0	0	0
2	2	0	1	0	0
3	3	0	0	1	0
4	4	0	0	0	1

# PMML 4.4 Models

- **Anomaly Detection (new)**
- **Association Rules Model**
- **Clustering Model**
- **General Regression**
- **Naïve Bayes**
- **Nearest Neighbor Model**
- **Neural Network**
- **Regression**
- **Tree Model**
- **Baseline Model**
- **Bayesian Network**
- **Gaussian Process**
- **Ruleset**
- **Scorecard**
- **Sequence Model**
- **Support Vector Machine**
- **Time Series**
- **Mining Model:** composition or ensemble (or both) of models



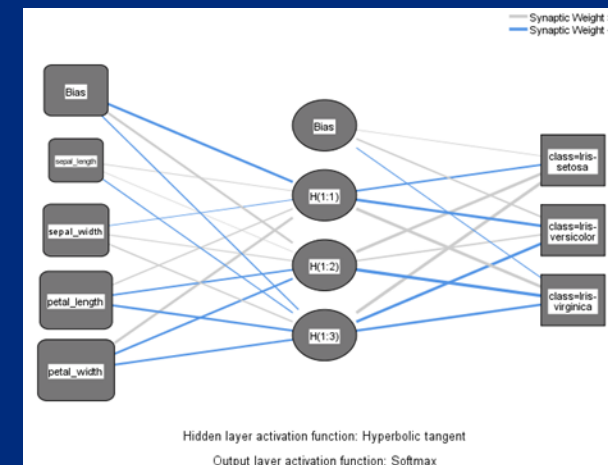
# Contents of a PMML Model

- ❖ **Mining Schema**: target and predictors, importance, missing value treatment, invalid value treatment, outlier treatment
- ❖ **Output**: what to report, post-processing
- ❖ **Model Stats**: description of input data
- ❖ **Model Explanation**: model diagnostics, useful for visualization
- ❖ **Targets**: target category info and prior probabilities
- ❖ **Local Transformations**: predictor transformations local to the model
- ❖ ...<Specific model contents>...
- ❖ **Model Verification**: expected results for some cases



# An example PMML – Data Dictionary, Transformations

```
▼<DataDictionary numberOfFields="5">
  ▼<DataField name="class" optype="categorical" dataType="string">
    <Value value="Iris-setosa"/>
    <Value value="Iris-versicolor"/>
    <Value value="Iris-virginica"/>
  </DataField>
  <DataField name="sepal_length" optype="continuous" dataType="double"/>
  <DataField name="sepal_width" optype="continuous" dataType="double"/>
  <DataField name="petal_length" optype="continuous" dataType="double"/>
  <DataField name="petal_width" optype="continuous" dataType="double"/>
</DataDictionary>
```

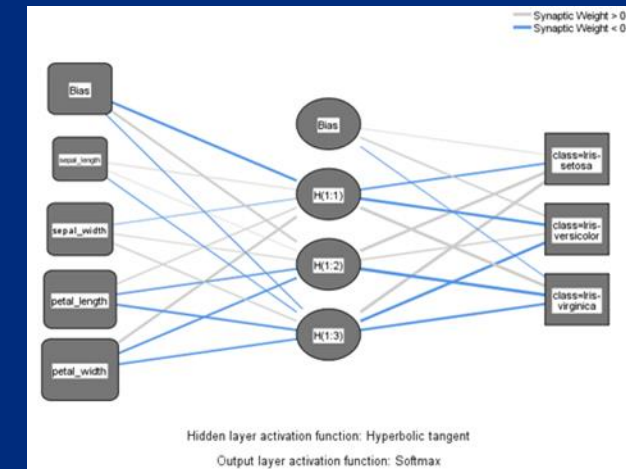


```
▼<DerivedField optype="categorical" dataType="double" name="classValue2">
  <NormDiscrete field="class" value="Iris-virginica"/>
</DerivedField>
▼<DerivedField optype="continuous" dataType="double" name="sepal_lengthNorm">
  ▼<NormContinuous field="sepal_length">
    <LinearNorm orig="4.3" norm="-1.84285714285714"/>
    <LinearNorm orig="7.7" norm="2.3204081632653"/>
  </NormContinuous>
</DerivedField>
▼<DerivedField optype="continuous" dataType="double" name="sepal_widthNorm">
  ▼<NormContinuous field="sepal_width">
    <LinearNorm orig="2" norm="-2.48539690378995"/>
    <LinearNorm orig="4.4" norm="3.13131926296699"/>
  </NormContinuous>
</DerivedField>
```

# Example PMML – Neural Network MiningSchema and inputs

```
▼<NeuralNetwork functionName="classification" activationFunction="tanh">
  ▼<MiningSchema>
    <MiningField name="sepal_length"/>
    <MiningField name="sepal_width"/>
    <MiningField name="petal_length"/>
    <MiningField name="petal_width"/>
    <MiningField name="class" usageType="predicted"/>
  </MiningSchema>
  ▼<NeuralInputs>
    ▼<NeuralInput id="0">
      ▼<DerivedField optype="continuous" dataType="double">
        <FieldRef field="sepal_lengthNorm"/>
      </DerivedField>
    </NeuralInput>
    ▼<NeuralInput id="1">
      ▼<DerivedField optype="continuous" dataType="double">
        <FieldRef field="sepal_widthNorm"/>
      </DerivedField>
    </NeuralInput>
```

*Predictors*



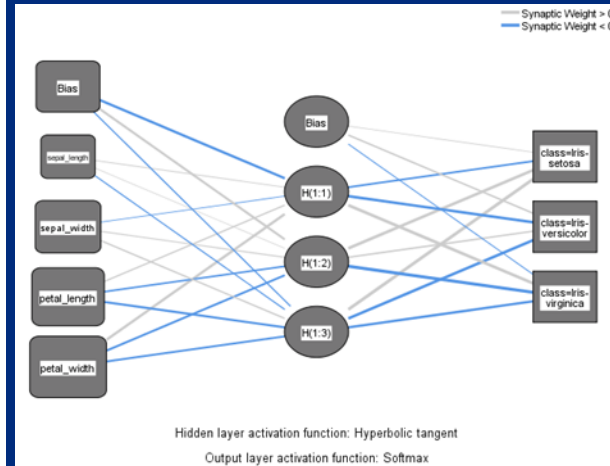
# Example PMML - Neural Network hidden layer and outputs

```
▼<Neuron id="6" bias="-0.69138649428932">
  <Con from="0" weight="-0.57324998362272"/>
  <Con from="1" weight="0.892806772564007"/>
  <Con from="2" weight="-1.23192787546061"/>
  <Con from="3" weight="-1.19705013526962"/>
</Neuron>
</NeuralLayer>
▼<NeuralLayer numberOfNeurons="3" activationFunction="identity" normalizationMethod="softmax">
  ▼<Neuron id="7" bias="0.101922887283541">
    <Con from="4" weight="-1.05690948855012"/>
    <Con from="5" weight="2.00228899161664"/>
    <Con from="6" weight="3.31278374396491"/>
  </Neuron>
  ▼<Neuron id="8" bias="0.917636281284728">
    <Con from="4" weight="-1.47230776836775"/>
    <Con from="5" weight="0.905795272070893"/>
    <Con from="6" weight="-1.60793177845373"/>
  </Neuron>
  ▼<Neuron id="9" bias="-0.2772471777484">
    <Con from="4" weight="2.22290439134024"/>
    <Con from="5" weight="-2.43960637239511"/>
    <Con from="6" weight="-1.32214182019044"/>
  </Neuron>
</NeuralLayer>
▼<NeuralOutputs>
  ▼<NeuralOutput outputNeuron="7">
    ▼<DerivedField optype="categorical" dataType="double">
      <FieldRef field="classValue0"/>
    </DerivedField>
  </NeuralOutput>
  ▼<NeuralOutput outputNeuron="8">
    ▼<DerivedField optype="categorical" dataType="double">
      <FieldRef field="classValue1"/>
    </DerivedField>
  </NeuralOutput>
  ▼<NeuralOutput outputNeuron="9">
    ▼<DerivedField optype="categorical" dataType="double">
      <FieldRef field="classValue2"/>
    </DerivedField>
  </NeuralOutput>
</NeuralOutputs>
```

*Hidden layer neuron*

*Output  
Layer  
Neurons*

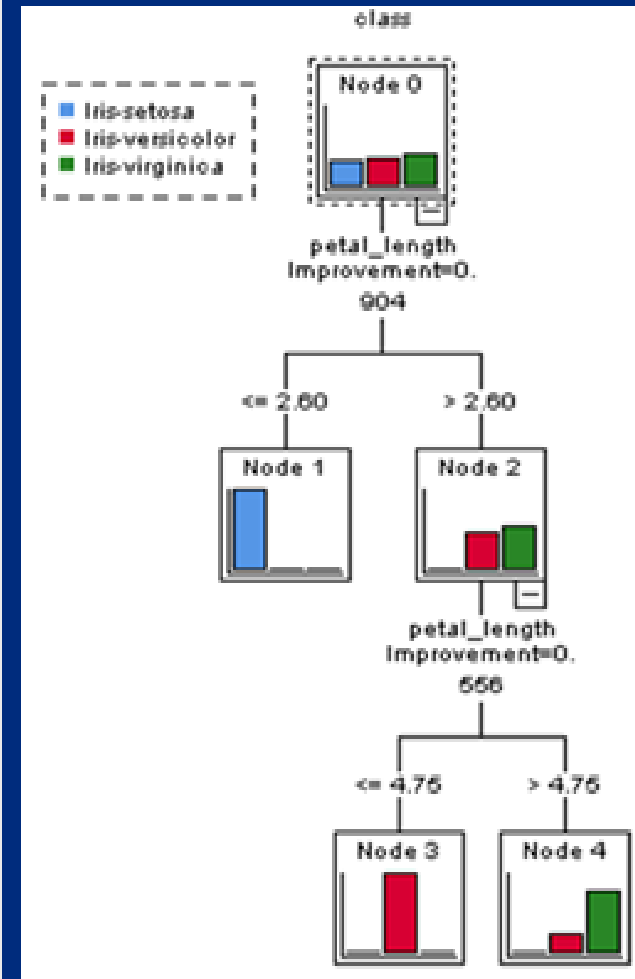
*Connecting  
target to the  
neurons*





# Example PMML for a Tree Model

```
<Node id="0"> <True/>
  <Node id="1" score="Iris-setosa" recordCount="50.0">
    <SimplePredicate field="petal_length" operator="lessOrEqual"
      value="2.6"/>
    <ScoreDistribution value="Iris-setosa" recordCount="50.0"/>
    <ScoreDistribution value="Iris-versicolor" recordCount="0.0"/>
    <ScoreDistribution value="Iris-virginica" recordCount="0.0"/>
  </Node>
  <Node id="2">
    <SimplePredicate field="petal_length" operator="greaterThan"
      value="2.6"/>
    <Node id="3" score="Iris-versicolor" recordCount="40.0">
      <SimplePredicate field="petal_length"
        operator="lessOrEqual" value="4.75"/>
```



# PMML Powered

From

<http://dmg.org/pmml/products.html>:

Alpine Data

Angoss

BigML

Equifax

Experian

FICO

Fiserv

Frontline Solvers

GDS Link

IBM (Includes SPSS)

IBM Developer

JPMML

KNIME

KXEN

Liga Data

Microsoft

MicroStrategy

NG Data

Open Data

Opera

Pega

Pervasive Data Rush

Predixion Software

Rapid I

R

Salford Systems (Minitab)

SAND

SAS

Software AG (incl. Zementis)

Spark

Sparkling Logic

Teradata

TIBCO

WEKA





# Agenda

- Challenges
- PMML Internals
- **PMML in Python and R**
- PMML in IBM products
- PFA
- ONNX

# PMML in Python

JPMML package is created and maintained by Villu Ruusmann in Estonia.

From <https://stackoverflow.com/questions/33221331/export-python-scikit-learn-models-into-pmml>

```
pip install git+https://github.com/jpmml/sklearn2pmml.git
```

***Example of how to export a classifier tree to PMML. First grow the tree:***

# example tree & viz from <http://scikit-learn.org/stable/modules/tree.html>

```
from sklearn import datasets, tree
```

```
iris = datasets.load_iris()
```

```
clf = tree.DecisionTreeClassifier()
```

```
clf = clf.fit(iris.data, iris.target)
```

*SkLearn2PMML conversion takes 2 arguments: an estimator (our `clf`) and a mapper for preprocessing. Our mapper is pretty basic, since no transformations.*

```
from sklearn_pandas import DataFrameMapper
```

```
default_mapper = DataFrameMapper([(i, None) for i in iris.feature_names + ['Species']])
```

```
from sklearn2pmml import sklearn2pmml
```

```
sklearn2pmml(estimator=clf, mapper=default_mapper, pmml="IrisClassificationTree.pmml")
```

# PMML in R

## R packages “pmml” and “pmm1Transformations”

<https://cran.r-project.org/package=pmml>

Supports a number of R models: `ada`, `amap`, `arules`, `caret`, `clue`, `data.table`, `gbm`, `glmnet`, `neighbr`, `nnet`, `rpart`, `randomForest`, `kernlab`, `e1071`, `testthat`, `survival`, `xgboost`, `knitr`, `rmarkdown`

Maintained by Dmitriy Bolotov and others from Software AG

JPMML also has package “**r2pmml**” that augments “pmml” and provides PMML export for additional R models

### Build and save a decision tree (C&RT) model predicting Species class:

```
> irisTree <- rpart( Species~., iris )  
> saveXML( pmml( irisTree ), "IrisTree.xml" )
```





# Agenda

- Challenges
- PMML Internals
- PMML in Python and R
- **PMML in IBM products**
- PFA
- ONNX

# IBM SPSS Statistics

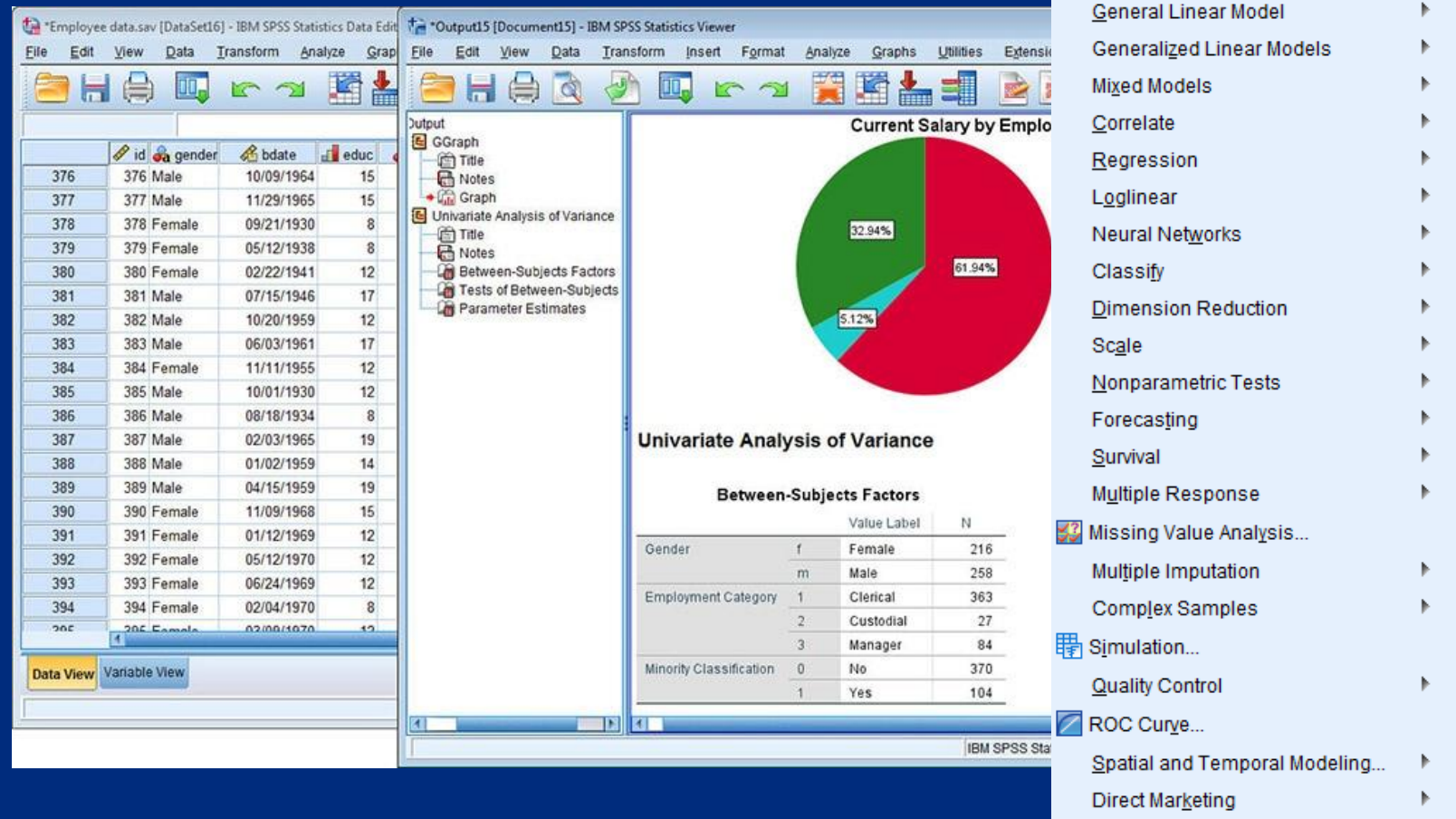
1968  
Statistical Package for Social Sciences

Acquired by IBM in 2009

Release 25 in August 2017,  
26 in Spring 2019.  
Subscription option

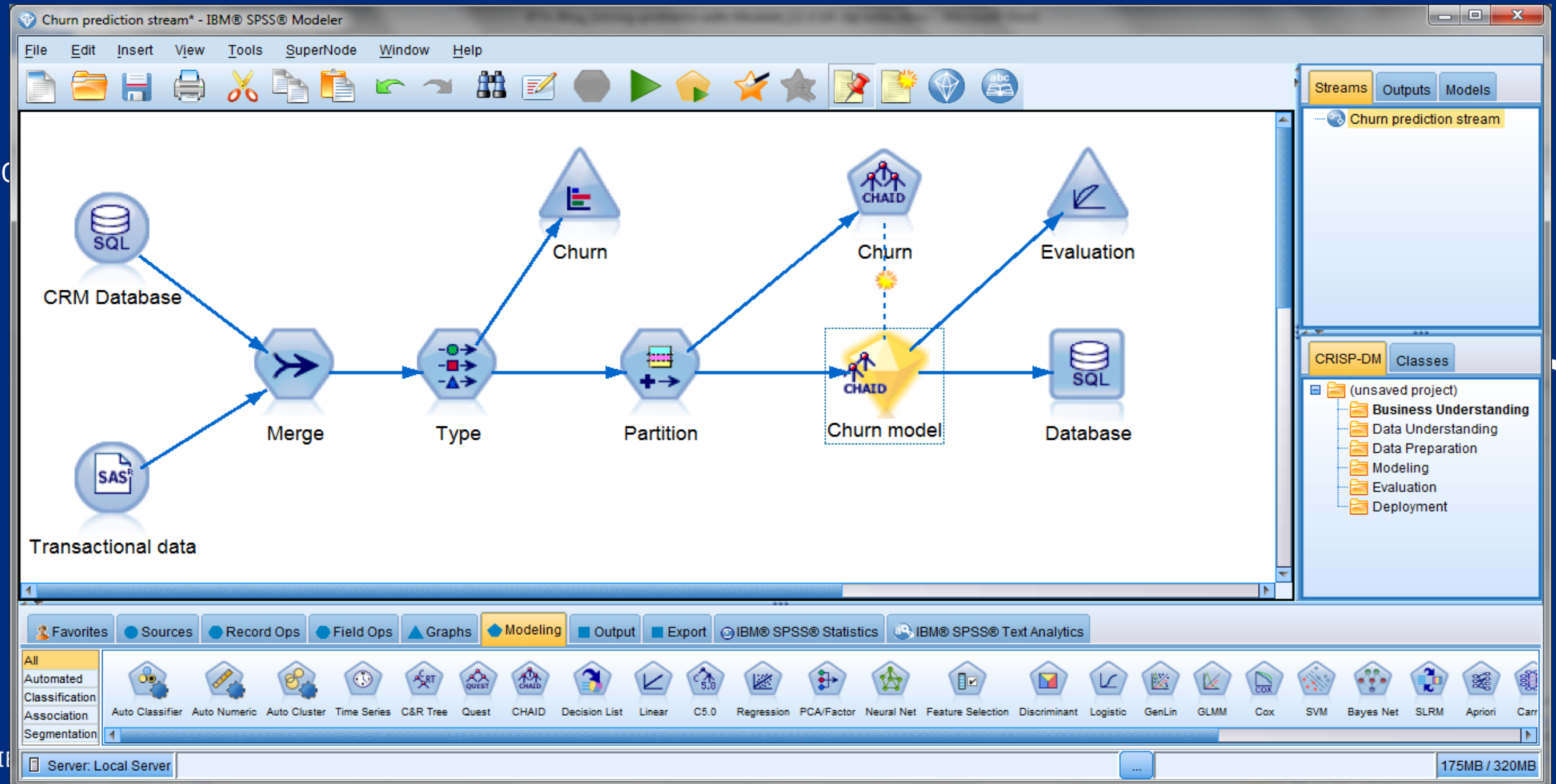
Integration with  
Python and R

IBM Developer





# IBM SPSS Modeler



# IBM SPSS Statistics

*Transformation PMML from:*

ADP (Automatic Data Preparation)

TMS Begin/TMS End

*Model PMML from:*

COXREG, CSCOXREG

CSGLM, CSLOGISTIC, CSORDINAL

GENLIN, Logistic regression, NOMREG

GENLINMIXED

LINEAR, KNN

MLP, RBF neural networks

NAÏVE BAYES

REGRESSION

TREE, TSMODEL

TWOSTEP CLUSTER

# IBM SPSS Modeler

Apriori, CARMA, Association Rules

C5, CART, Chaid decision trees

Cox regression

GENLIN

Decision List

K-Means Cluster

KNN

LINEAR, Regression

Logistic Regression

MLP and RBF

NOMREG

Random Trees

Regression

Two Step Cluster

# Score PMML in IBM SPSS Statistics

Utilities->Scoring Wizard

Scoring Wizard

Select a Scoring Model:

BayesRegrServo.xml  
Employee\_Chaid.xml  
rbf\_pmml.xml  
tree\_1.0-37.zip  
TreeServo\_PMML.xml

Model Details:

ModelMethod : TREE\_PMML  
Ensemble Method : none  
Application : IBM SPSS Statistics 25.0.0.0  
Target : jobcat  
Split :  
Predictor : gender, salary

Browse...

C:\Users\slevitan\Documents\Advanced...

< Back

Next >

Finish

Cancel

Help

Scoring Wizard

Model Name: Employee\_Chaid  
Model Type: TREE\_PMML

Select Scoring Functions:  
Each selected function will create a new field in the dataset.

	Function	Field Names	Value
<input checked="" type="checkbox"/>	Probability of Predicted Category	PredictedProbability	
<input checked="" type="checkbox"/>	Probability of Selected Category	SelectedProbability	3
<input checked="" type="checkbox"/>	Predicted Value	PredictedValueFromPMML	
<input checked="" type="checkbox"/>	Node Number	NodeNumber	
<input checked="" type="checkbox"/>	Confidence	Confidence	

Scoring Wizard

Model Name: IrisTree  
Model Type: Tree

Match Model Fields to the Dataset:

Dataset Fields	Model Fields	Role	Measure	Type
<input checked="" type="checkbox"/> sepal_length	Sepal.Length	Predictor	Continuous	Numeric
<input checked="" type="checkbox"/> sepal_width	Sepal.Width	Predictor	Continuous	Numeric
<input checked="" type="checkbox"/> petal_length	Petal.Length	Predictor	Continuous	Numeric
<input checked="" type="checkbox"/> petal_width	Petal.Width	Predictor	Continuous	Numeric
<input checked="" type="checkbox"/> class				

Missing Values

☒ Use value substitution  
☐ Use system-missing

< Back

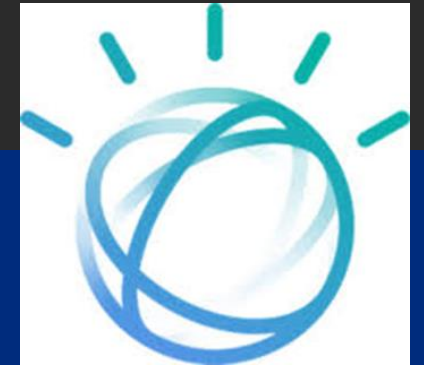
Next >

Finish

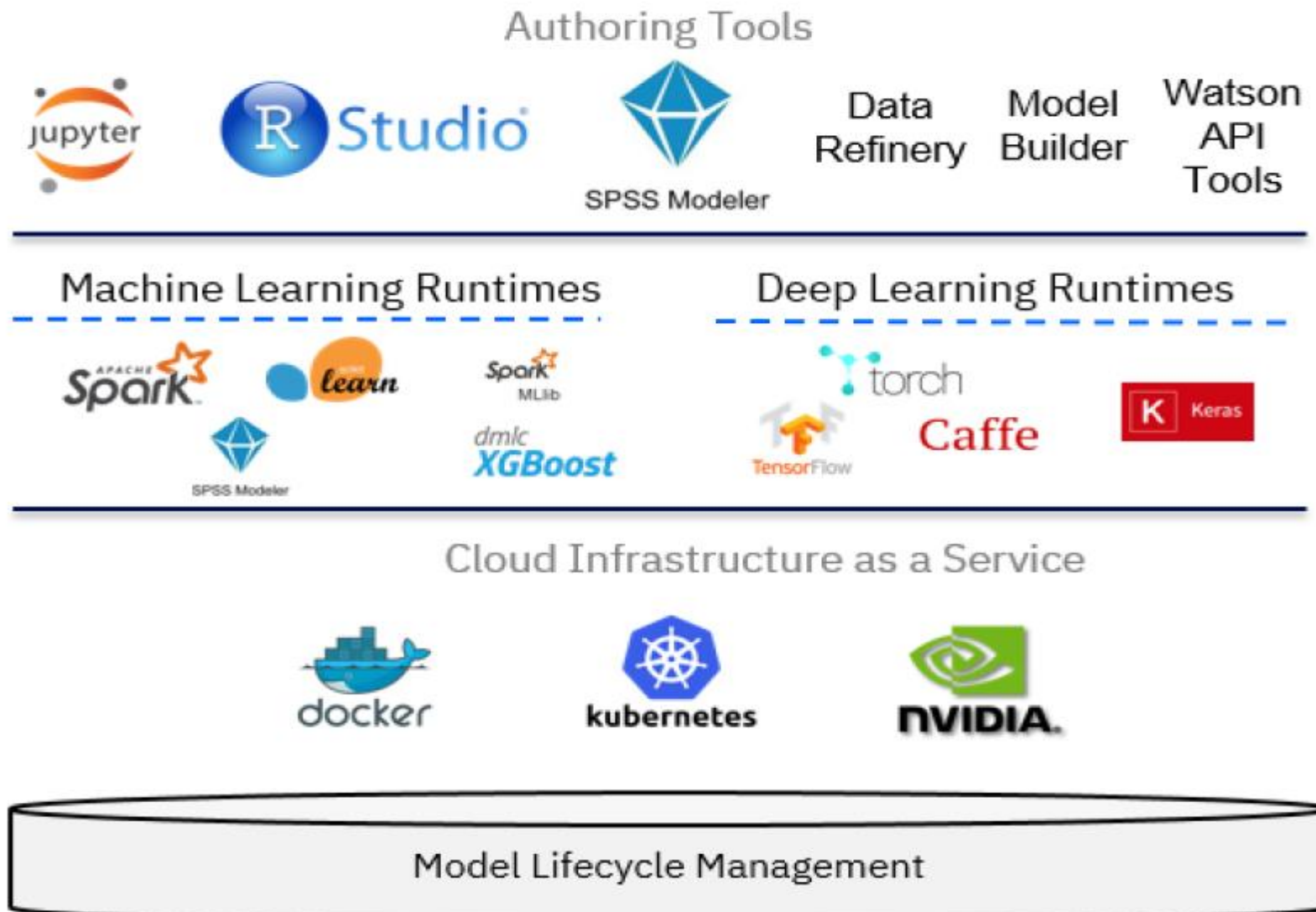
Cancel

Help

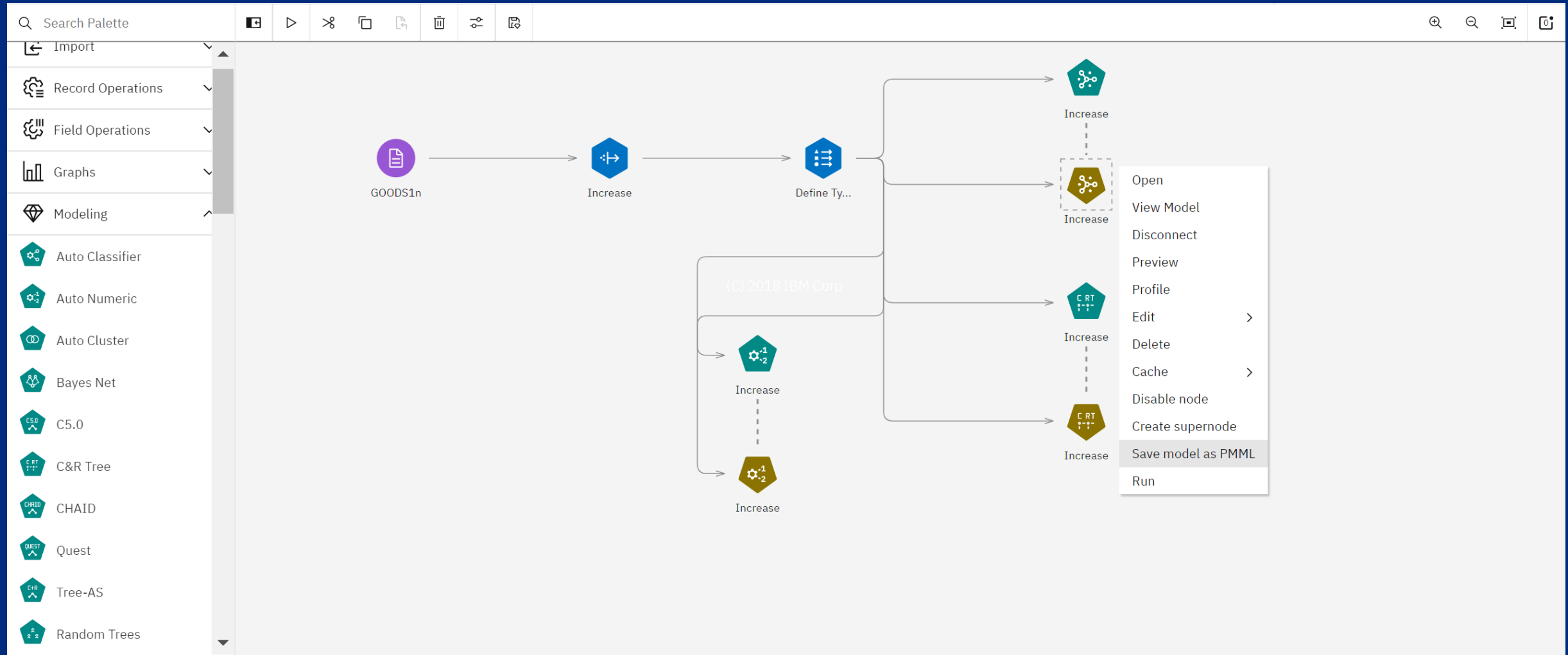
# Watson Studio (formerly Data Science Experience)



PMML export possible in Jupyter notebooks, Modeler flows, R Studio. PMML scoring can be done in Flows, notebooks, Watson Machine Learning.



# Watson Studio Flows on IBM Cloud. Free account: <https://ibm.biz/Bdqie5>



# Scoring PMML in Watson Machine Learning

The screenshot shows the 'PMML Scorer' interface in the IBM Watson console. The 'Overview' tab is selected, displaying a summary of the model deployment. Below the summary, the 'Input Schema' is shown as a table with columns for feature names and their data types.

Machine learning service	pm-20-dx
Model Type	pmml-4.3
Runtime environment	java-1.8
Training date	11 Sep 2018, 4:24 PM
Label column	Species
Latest version	fc1f4805-58e0-4130-a91f-8f1cbd3f0e50

COLUMN	TYPE
Sepal.Length	double
Sepal.Width	double
Petal.Length	double
Petal.Width	double

The screenshot shows the 'Implementation' tab of the 'DeployIrisTree' model. It provides the Scoring End-point, Authorization details, and Content-type. Below this, the 'Code Snippets' section shows a Python code snippet for making a REST API call to score the model.

**Scoring End-point:** `https://us-south.mlcloud.ibm.com/v3/wmldeployments/4fa4556f-fa...`

**Authorization:** Bearer <token> (See code snippets below for information on how to retrieve the token.)

**Content-type:** application/json (Required if the request body is sent in JSON format.)

**Code Snippets:**

```
import urllib3, requests, json

# retrieve your wml_service_credentials_username, wml_service_credentials_password
# Service credentials associated with your IBM Cloud Watson Machine Learning

wml_credentials={
    "url": wml_service_credentials_url,
    "username": wml_service_credentials_username,
    "password": wml_service_credentials_password
}

headers = urllib3.util.make_headers(basic_auth='{username}:{password}'.format(
    username=wml_credentials['username'], password=wml_credentials['password']))
url = '{}/v3/identity/token'.format(wml_credentials['url'])
response = requests.get(url, headers=headers)
mltoken = json.loads(response.text).get('token')
```

The screenshot shows the 'Test' tab of the 'DeployIrisTree' model. It allows users to enter input data for the model and click the 'Predict' button to get the output. The input fields are for Sepal.Length, Sepal.Width, Petal.Length, and Petal.Width, each with a value of 4. The output shows the predicted value and probability for the 'virginica' species.

**Enter input data:**

Sepal.Length: 4  
Sepal.Width: 4  
Petal.Length: 4  
Petal.Width: 4

**Predict**

**Output:**

```
{
  "fields": [
    "PredictedValue",
    "Probability",
    "Probability"
  ],
  "values": [
    "virginica",
    0,
    0.021739130434782608,
    0.9782608695652174
  ]
}
```



# Benefits of PMML



Allows  
seamless  
deployment  
and model  
exchange

Transparency:  
human and  
machine-  
readable

Fosters best  
practices in  
model  
building and  
deployment





# Agenda

- Challenges
- PMML Internals
- PMML in Python and R
- PMML in IBM products
- **PFA**
- ONNX

# Portable Format for Analytics - PFA

PMML is great, except when a model or feature is not supported

PFA to overcome this

JSON format, AVRO schemas for data types

A mini functional math language + schema specification

Built-in functions and simple models.

Info: [dmg.org/pfa](https://dmg.org/pfa)



Jim Pivarski

# A Simple Example of PFA (copied from Nick Pentreath's presentation)

- Example – multi-class logistic regression
- Specify input and output types using Avro schemas

```
{
  "name": "logistic-regression-model",
  "input": {
    "type": {
      "type": "array",
      "items": "double"
    }
  },
  "output": {
    "type": "double"
  },
}
```

- Specify the *action* to perform (typically on input)

```
"action": [
  {
    "a.argmax": [
      {
        "m.link.softmax": [
          {
            "model.reg.linear": [
              "input",
              {
                "cell": "model"
              }
            ]
          }
        ]
      }
    ]
  }
]
```

# Known Support for PFA

**Hadrian** (PFA export and scoring engine)  
from Open Data Group (Chicago, IL)



**Aardpfark** (PFA export in SparkML)  
by Nick Pentreath, IBM CODAIT, South Africa

**Woken** (PFA export and validation)  
by Ludovic Claude, CHUV, Lausanne, Switzerland



There was a lot of interest in PFA.

Many opportunities for open source contributions.



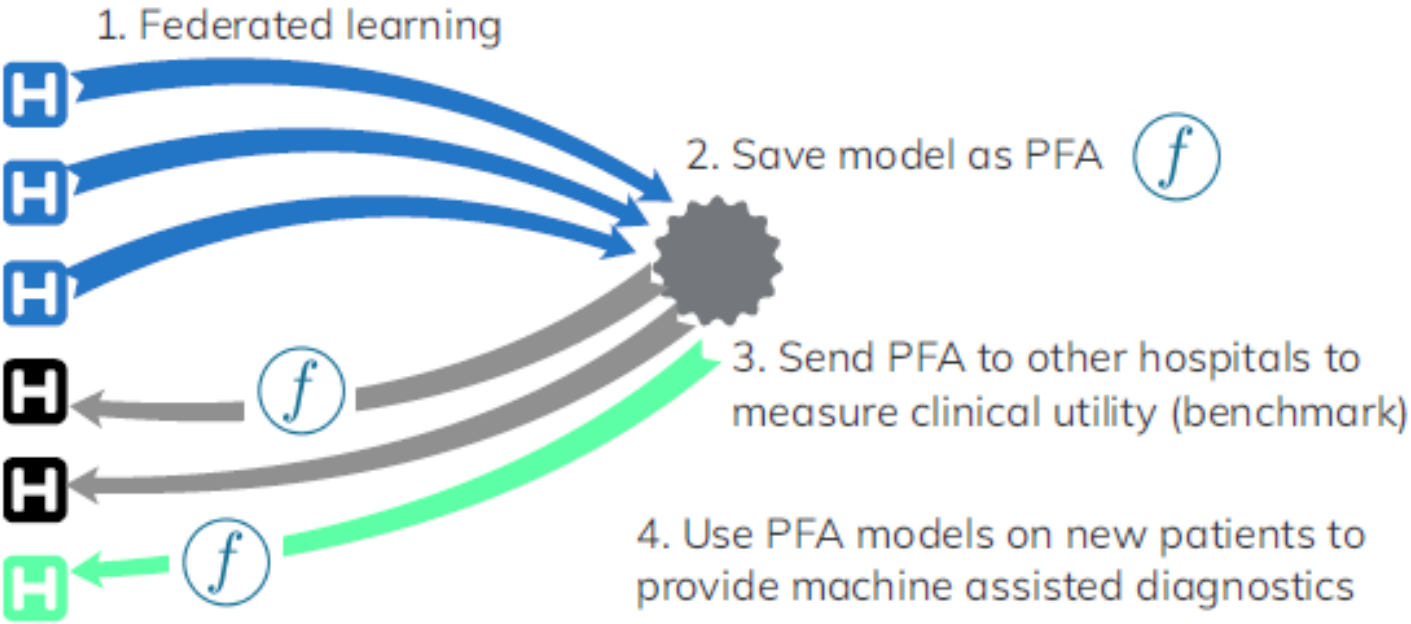
# Use of PMML and PFA in medical applications

## Human Brain Project

Ludovic Claude,  
CHUV  
Lausanne,  
Switzerland



### MIP network of hospitals





# Agenda

- Challenges
- PMML Internals
- PMML in Python and R
- PMML in IBM products
- PFA
- **ONNX**

# ONNX: Open Neural Network eXchange



Since Sep. 2017. Protobuf

Covers DL and traditional ML

Active work by many companies



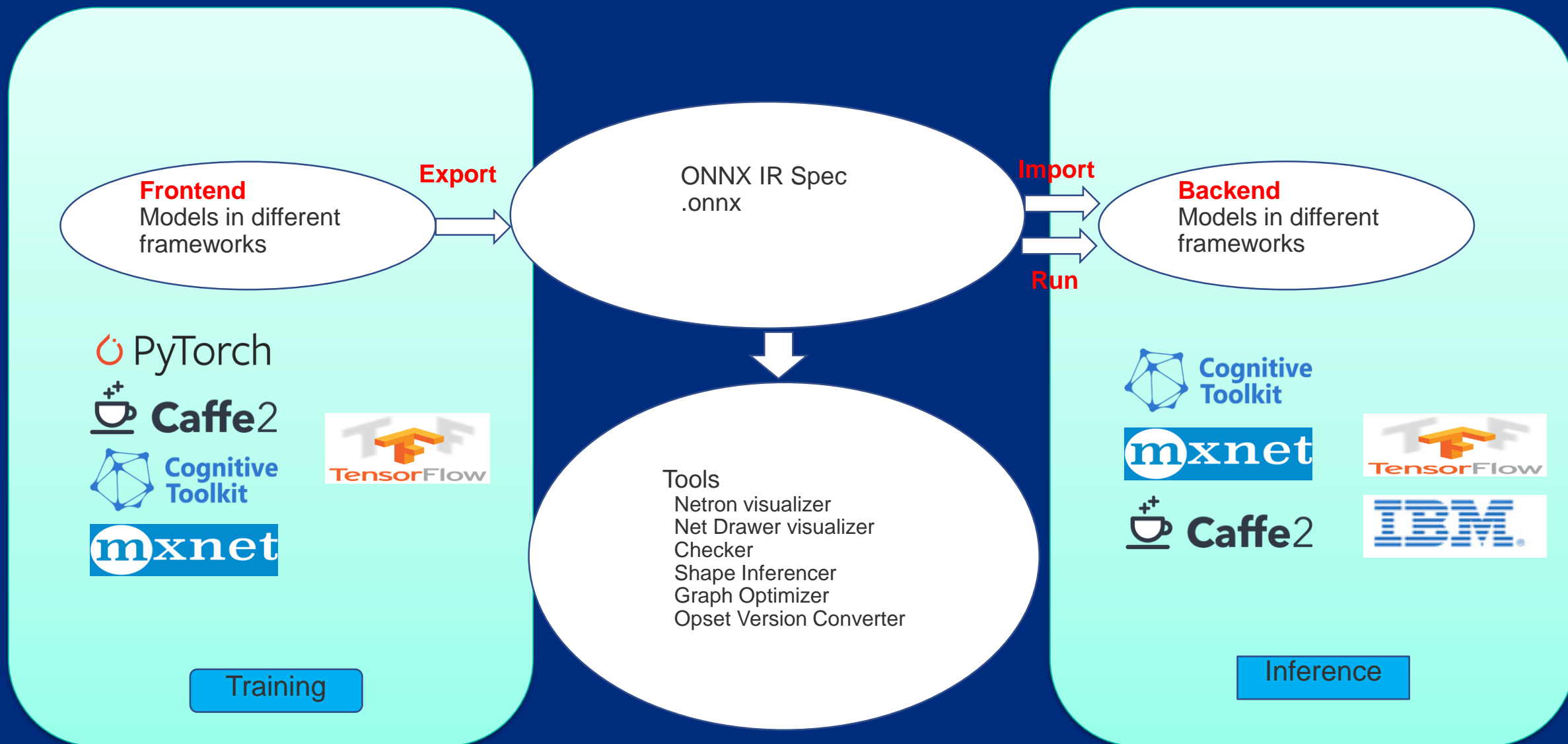


# ONNX Background



- Initial goal: make it easier to exchange trained models between DL frameworks.
- ONNX github has 24 repos, onnx is the core. Others are tutorials, model zoo, importers and exporters for frameworks.
- Onnx/onnx currently has 14 releases, 154 contributors, 8.1K stars.
- Release 1.7 expected March 31, 2020
- Core is in C++ with Python API and tools.
- Supported frameworks: Caffe2, Chainer, Cognitive Toolkit (CNTK), Core ML, MXNet, PyTorch, PaddlePaddle; TF in progress

# ONNX use pattern (diagram by Chin Huang)





# ONNX tutorials: import and export from frameworks



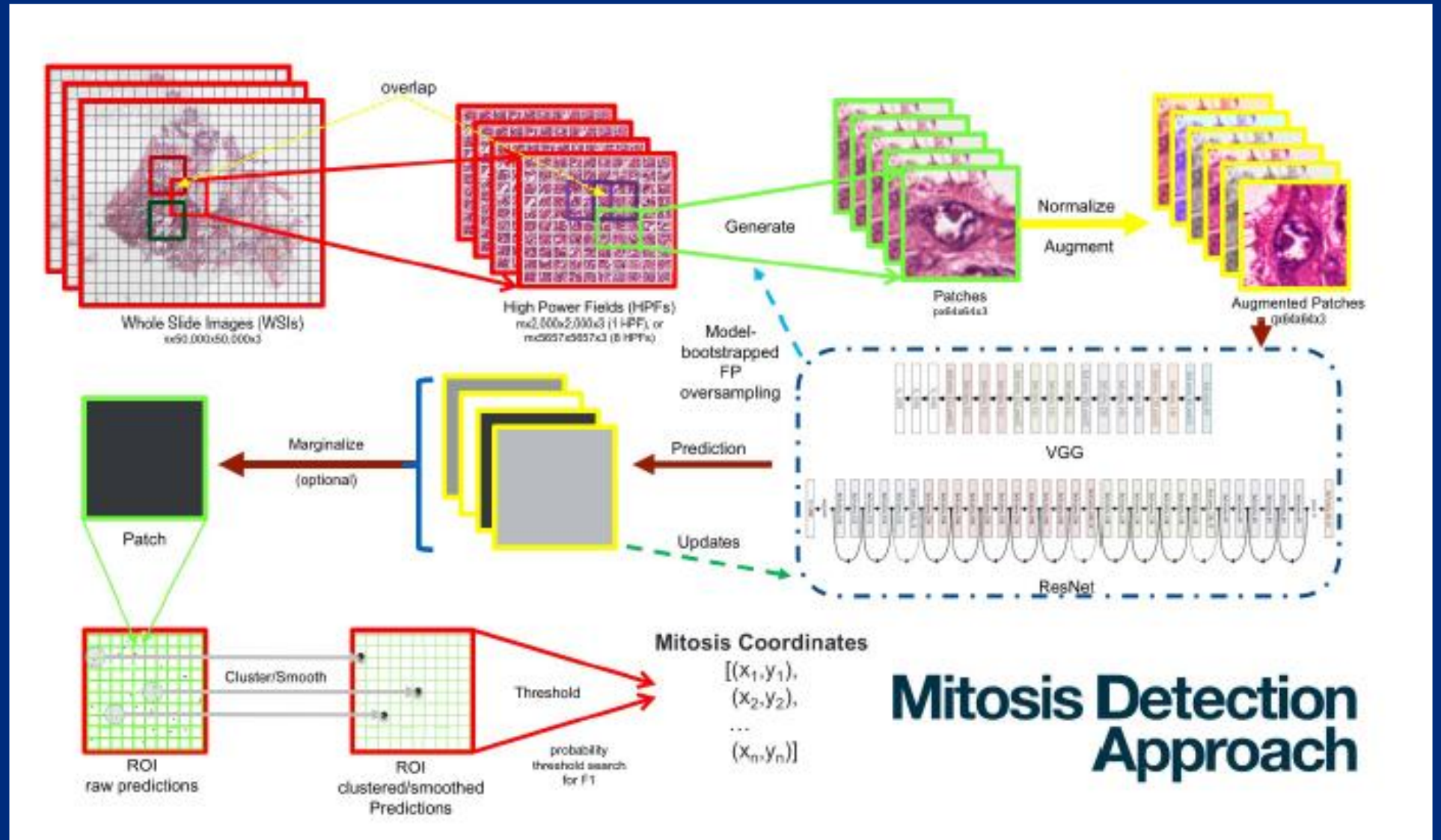
Framework / tool	Installation	Exporting to ONNX (frontend)	Importing ONNX models (backend)
Caffe	<a href="#">apple/coremltools</a> and <a href="#">onnx/onnxmltools</a>	Exporting	n/a
Caffe2	<a href="#">part of caffe2 package</a>	Exporting	Importing
Chainer	<a href="#">chainer/onnx-chainer</a>	Exporting	coming soon
Cognitive Toolkit (CNTK)	built-in	Exporting	Importing
Apple CoreML	<a href="#">onnx/onnx-coreml</a> and <a href="#">onnx/onnxmltools</a>	Exporting	Importing
Keras	<a href="#">onnx/keras-onnx</a>	Exporting	n/a
LibSVM	<a href="#">onnx/onnxmltools</a>	Exporting	n/a
LightGBM	<a href="#">onnx/onnxmltools</a>	Exporting	n/a
MATLAB	<a href="#">onnx converter on matlab central file exchange</a>	Exporting	Importing
Menoh	<a href="#">pfnet-research/menoh</a>	n/a	Importing
ML.NET	built-in	Exporting	Importing
Apache MXNet	<a href="#">part of mxnet package docs github</a>	Exporting	Importing
PyTorch	<a href="#">part of pytorch package</a>	Exporting, Extending support	coming soon
SciKit-Learn	<a href="#">onnx/sklearn-onnx</a>	Exporting	n/a
TensorFlow	<a href="#">onnx/onnx-tensorflow</a> and <a href="#">onnx/tensorflow-onnx</a>	Exporting - ONNX-Tensorflow Exporting - Tensorflow-ONNX	Importing [experimental]
TensorRT	<a href="#">onnx/onnx-tensorrt</a>	n/a	Importing



# Using ONNX in medical image processing: potential applications

MAX = Model  
Asset eXchange

[ibm.biz/  
model-exchange](https://ibm.biz/model-exchange)



# Conclusions



Model deployment is an important part of ML lifecycle

DMG works on open standards for model deployment

PMML eases deployment for supported models and data prep

ONNX is a de-facto standard for Deep Learning

Many opportunities for open source contributions



## Participants:

Register for the challenge, and start building

## Sponsors:

Show your full support with a sponsorship



Call for Code  
Founding Partner



Call for Code  
Creator



Call for Code  
Charitable Partner



Call for Code  
Affiliate

Join a **movement** of:

- **210,000** developers, data scientists & problem solvers
- **165+** nations
- **8,000+** applications built in the prior 2 years

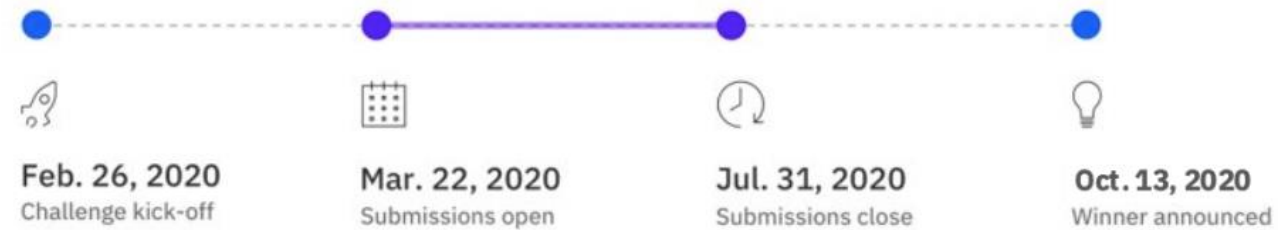
In line with the UN 75<sup>th</sup> Anniversary global conversation, help halt & reverse climate change by addressing:

- **Water sustainability**
- **Energy sustainability**
- **Disaster resiliency**

Have the chance to be **awarded:**

- **\$200,000** USD
- Open Source support from **The Linux Foundation**
- Meetings with **mentors** & potential **investors**
- **Solution implementation** support through **Code and Response™**

### Challenge timeline



[ibm.biz/callforcode](https://ibm.biz/callforcode)



# Links and resources

@SvetaLevitan

PMML [dmg.org/pmml](http://dmg.org/pmml)

Call for Code: [ibm.biz/callforcode](http://ibm.biz/callforcode)

PFA [dmg.org/pfa](http://dmg.org/pfa)

ONNX [onnx.ai](http://onnx.ai), [gitter.im/onnx](https://gitter.im/onnx)

Upcoming ONNX meeting on April 9 at 9am-12pm Pacific time:

<https://events.linuxfoundation.org/lf-ai-day-onnx-community-virtual-meetup/>

SPSS: <https://www.ibm.com/analytics/spss-statistics-software>

Watson Studio: <https://www.ibm.com/cloud/watson-studio>

Sign up for free IBM Cloud account: **<https://ibm.biz/Bdqie5>**

Join Chicago Meetups: Big Data Developers in Chicago, Chicago ML, ChiPy, ...

Replays of Virtual community day “Deploy AI” from March 10:

<https://ibm-deployai.bemyapp.com/>





Thank you.