What’s cooking

Bernd Wiswedel
Outline

Continued development of all products, including

• KNIME Server
• KNIME Analytics Platform

• KNIME Big Data Extensions
  (discussed after this session)
KNIME Server – What’s cooking

• Full server functionality via REST
• Admin Portal
• Distributed workflow executor (on Hadoop, ...)
• Web Portal: Guided Analytics

• Social Workflow Repository
• PMML Scoring as a web service
Social Workflow Repository

- Patrick Winter -
- Alexander Fillbrunn -

de.NBI
Workflow Sharing Made Easy

- Share, rate and discuss workflows on your KNIME Server
- Instantly updated when new workflow is uploaded
- Supports snapshot versioning
- Responsive design → Works on all device types
The Start Page

- Configurable lead text and news segment
- Most recent, top rated and random workflows
The Workflow Search

- Search workflows by title or author
- Sort by title, author, rating or upload date
Workflow Details

- Workflow description, image, rating and comments
- List of required plugins
- Download of all available workflow versions
Social Features

- Rating: 1 to 5 stars
- Comments
- Markdown formatting supported
- Release of new versions included in the comment timeline
PMML Scoring as a web service

- Alexander Fillbrunn -
PMML Compilation & Scoring

• Last year, we demonstrated the PMML to Java compiler
• Translates PMML to Java Virtual Machine Bytecode
• Can be used for faster scoring
• Now we make those compiled models much more powerful
How can we reuse the model we just learned?
Compiled Model to Jar Writer

- Bytecode is exported as a Java Archive
- Includes custom metadata
  - Lets other software know which class is the entry point
- Has no dependencies
  - Run it anywhere, as long as a Java Virtual Machine is available

Export model as JAR → mymodel.jar (4 KB) → Database
Compiled Models on KNIME Server

• Upload models to KNIME Server
• Immediately available as a RESTful Web Service
  – Update & Delete models
  – Retrieve model metadata
    • Model type, Input & output fields
  – Execution
• Data transferred via JSON
• Web Interface for management and testing
• Compatible with KNIME PMML models
Using Remote Compiled Models in KNIME

• Connects to any KNIME Server via REST interface
• Uploads the model’s bytecode
• Works for predictive models and transformations
Using Remote Compiled Models in KNIME

• Models callable by an identifier set in the PMML Compiler
• Appends results to the original table
Model Management on KNIME Server

- Web management interface
- Responsive design
  - Runs on Smartphones, Tablets and PCs
- Allows uploading, deleting and execution of models
KNIME Analytics Platform – What’s cooking

Enhanced analytics modules:

• Deeplearning4j integration

• Trees and tree ensembles
Deeplearning4j

• “Deeplearning4j is the first commercial-grade, open-source, distributed deep-learning library written for Java and Scala” [deeplearning4j.org]
• GPU integration
• Scalable on Hadoop, Spark
Deep Learning - Applications

• Face Recognition
  (97.5%) Accuracy
Deep Learning - Applications

- Image Recognition (83% Top-5 Accuracy 1000 classes!)
Deeplearning4j - Workflow
Deeplearning4j - Workflow
Deeplearning4j - Example Networks

Deep Convolutional Network (LeNet)

Deepbelief Network
Easy network architecture design
• Modular
  – Layerwise design of networks
• Model Import/Export
  – Caffe Import
• Beginner friendly
  – Import pretrained networks
• Highly configurable
Trees and Tree Ensembles

- Adrian Nembach-
Full workflow
**Input table**

![Table Image](image-url)
Tree Ensemble Dialog
Similarity Search Dialog
Nearest Neighbors
Nearest Neighbors

<table>
<thead>
<tr>
<th>Row ID</th>
<th>Image</th>
<th>zebra</th>
<th>distance</th>
<th>Image (4x4)</th>
<th>zebra</th>
</tr>
</thead>
<tbody>
<tr>
<td>raw2013_12_1</td>
<td><img src="zebra1.png" alt="Image" /></td>
<td>0.3%</td>
<td></td>
<td><img src="zebra2.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>raw2013_12_2</td>
<td><img src="zebra3.png" alt="Image" /></td>
<td>0.3%</td>
<td></td>
<td><img src="zebra4.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>raw2013_12_3</td>
<td><img src="zebra5.png" alt="Image" /></td>
<td>0.7%</td>
<td></td>
<td><img src="zebra6.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>raw2013_12_4</td>
<td><img src="zebra7.png" alt="Image" /></td>
<td>0.8%</td>
<td></td>
<td><img src="zebra8.png" alt="Image" /></td>
<td></td>
</tr>
<tr>
<td>raw2013_12_5</td>
<td><img src="zebra9.png" alt="Image" /></td>
<td>0.8%</td>
<td></td>
<td><img src="zebra10.png" alt="Image" /></td>
<td></td>
</tr>
</tbody>
</table>
Gradient Boosting Dialog
# Gradient Boosting Results

![Confusion Matrix](image)

## Confusion Matrix - 0.39 - Scorer (score prediction)

<table>
<thead>
<tr>
<th></th>
<th>donkey</th>
<th>monkey</th>
<th>goat</th>
<th>wolf</th>
<th>jetski</th>
<th>zebra</th>
<th>centaur</th>
<th>mug</th>
<th>statue</th>
<th>building</th>
<th>bag</th>
<th>carriage</th>
</tr>
</thead>
<tbody>
<tr>
<td>donkey</td>
<td>12</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>monkey</td>
<td>0</td>
<td>41</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>goat</td>
<td>1</td>
<td>0</td>
<td>16</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>wolf</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>22</td>
<td>1</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>jetski</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>zebra</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>centaur</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>mug</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>statue</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>building</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>bag</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>carriage</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Correct classified:** 441  
**Wrong classified:** 88  
**Accuracy:** 83.365 %  
**Error:** 16.635 %  
**Cohen's kappa ($\kappa$):** 0.814
Trees and Tree Ensembles: Changes “under the hood”

- Support of binary splits for nominal attributes
- Missing value handling
- Support of byte vector data (high-dimension count fingerprints)
- Code optimization
  - Runtime
  - Memory
Trees and Tree Ensembles: New nodes

• Gradient Boosting
  – Also based on tree ensembles
  – Boosting: Improving an existing model by adding a new model
  – Shallow trees

• Random Forest Distance
  – Distance measure induced by a random forest
  – Based on proximity
Semantic Web Integration

- Access and manipulate semantic web resources e.g. DBpedia
- Execute semantic queries via SPARQL
- Usage model similar to database integration
KNIME Analytics Platform – What’s cooking

... there is more coming, including

• More preprocessing nodes (as needed)
• REST client nodes – forking KREST community extension
• More streaming-enabled nodes
• ...

© 2016 KNIME.com AG. All Rights Reserved.
The KNIME® trademark and logo and OPEN FOR INNOVATION® trademark are used by KNIME.com AG under license from KNIME GmbH, and are registered in the United States.