Guided Automation

Christian Dietz, Paolo Tamagnini

20 March 2019

#KNIMESummit2019
Automating What?

Automating Analytics:
- Parameter Optimization
- Model Selection
- Ensemble Construction
Automating What?

- Model Parameter Optimization

Automatically try a lot of model parameters.
Automating What?

- **Model Selection**
  - Try many models, but in an automated way.
Automating What?

• Model Selection and Parametrization

Check out Daria’s Blogpost on knime.com “Stuck in the Nine Circles of Hell? Try Parameter Optimization & A Cup Of Tea” (05/28/18)
Automating What?

Automating Data Proc:
- Feature Selection
- Feature Construction
- Data Cleaning

Automating Analytics:
- Parameter Optimization
- Model Selection
- Ensemble Construction
Automating What?

• Data Cleansing
  – Missing value handling, calculate statistics, outlier detection

• Feature Engineering
  – Mathematical transformations, feature combinations, dimensionality reduction and more

• Feature Selection
  – Forward feature selection, backward feature elimination, genetic algorithm etc.
Automating What?

Automating Data Integration:
- Parsing
- Record Matching
- ...

Automating Data Proc:
- Feature Selection
- Feature Construction
- Data Cleaning

Automating Analytics:
- Parameter Optimization
- Model Selection
- Ensemble Construction
Automating How?

• Try: (Brute Force)
  – Use compute power to try variations

• Copy: (Transfer Learning)
  – Use similar problems as starting point

• Learn: (Bayesian Optimization)
  – Use Machine Learning to predict/learn parameter space
Automating Everything?

Automating Data Proc:
- Feature Selection
- Feature Construction
- Data Cleaning

Automating Data Integration:
- Parsing
- Record Matching
- ...

Human Input Needed:
- Data Selection – Is this relevant?!
- Analysis Goal – What is interesting?
- Exploration – This looks weird?...

Automating Analytics:
- Parameter Optimization
- Model Selection
- Ensemble Construction
Automating How?

• Try: (Brute Force)
  – Use compute power to try variations
• Copy: (Transfer Learning)
  – Use similar problems as starting point
• Learn: (Bayesian Optimization)
  – Use Machine Learning to predict/learn parameter space
• Interact: (Guided Automation)
  – Augment Automation with Human Intelligence
Flexible Automation / Interaction

• Data Scientists build (or start from template)
  – automate the boring pieces
  – add interaction where human feedback is needed

• Deploy to Business Users as Analytical App
  – hides complexity
  – enables interaction at the right level of detail

(How much interaction? That depends...)

Building a Guided Automation Workflow

Interaction Points

Automated
KNIME’s Guided Automation: Automation + Interaction

Interaction Points

Automated
(prepackaged, updated, or created by in-house experts)

www.myserver.ch
Guided Automation: Automation + Interaction
Guided Automation: Automation + Interaction
Guided Automation: Automation + Interaction

Upload Data and Process Setup
1. Upload your data / define file path
2. Select the target column for the prediction
3. Filter columns to exclude from the model
4. Select models and whether to fine-tune the model parameters

Fine-tune Model Parameters
By default (unchecked), all parameters for the selected models and for feature engineering are automatically fully optimized. However, by checking this option you can guide the automatic optimization process.

✓ Finetune Model Parameters

Select Models
Guided Automation: Automation + Interaction

1. Split the data into training and testing data.
2. Preprocess the training data.
3. Model optimization. At first, the hyperparameters are optimized. After that, if enabled, feature engineering is performed using the optimized hyperparameters for the models.
4. Retrain the models with the optimized parameters and features using the whole training data.

Split data in training and testing.
Preprocess training data.
Optimize model parameters.
Optimize feature selection.
Retrain the optimized models and collect results.

Output:
1. Preprocessed training data
2. Preprocessing models
Output:
1. Models
2. Training data
Output:
1. Models
2. Training features
2. Predictions - testing data

Apply preprocessing on testing data.
Guided Automation: Automation + Interaction
Guided Automation: Automation + Interaction

• Live Demo
Scoring Workflow

1. **Table Reader**
   - Model parameters.

2. **Read data**
   - Filter selected features
   - Apply preprocessing on testing data

3. **Cell To Model**
   - Apply Feature Creation to Test Data
   - Logistic Regression Predictor

4. **Table View**
Customize the Blueprint for Text Processing

Added by in-house expert.
Guided Analytics and Automated AI, ML, ...

In a Nutshell:
- Automation can take out the drivers...
- ...but then it also takes away their expertise.

Guided Automation allows
- to automate the boring pieces...
- ...but keep the expert in the loop.

Integration matters. Nobody has all the pieces.
Guided Analytics for Machine Learning Automation

This workflow generates a fully automated web based application to select, train, test, and optimize a number of machine learning models. The workflow was designed for business analysts to easily create predictive analytics solutions by applying their domain knowledge. Each of the wrapped metanodes will generate a web page with which the business analyst can interact.
Upload Dataset

Upload the dataset to be used.

Uploading file "adult.csv"
Select Target

Select the target column whose values should be predicted.

Select:

<table>
<thead>
<tr>
<th>ID</th>
<th>Workclass</th>
<th>Education</th>
<th>Education-Num</th>
<th>Marital Status</th>
<th>Occupation</th>
<th>Relationship</th>
<th>Race</th>
<th>Sex</th>
<th>Country</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>State-gov</td>
<td>Bachelors</td>
<td>13</td>
<td>Never-married</td>
<td>Adm-clerical</td>
<td>Not-in-family</td>
<td>White</td>
<td>Male</td>
<td>United-States</td>
<td>poor</td>
</tr>
<tr>
<td>1</td>
<td>Self-emp-not-</td>
<td>Bachelors</td>
<td>13</td>
<td>Married-civ-spouse</td>
<td>Exec-managerial</td>
<td>Husband</td>
<td>White</td>
<td>Male</td>
<td>United-States</td>
<td>poor</td>
</tr>
<tr>
<td>2</td>
<td>Private</td>
<td>HS-grad</td>
<td>9</td>
<td>Divorced</td>
<td>Handlers-cleaners</td>
<td>Not-in-family</td>
<td>White</td>
<td>Male</td>
<td>United-States</td>
<td>poor</td>
</tr>
<tr>
<td>3</td>
<td>Private</td>
<td>11th</td>
<td>7</td>
<td>Married-civ-spouse</td>
<td>Handlers-cleaners</td>
<td>Husband</td>
<td>Black</td>
<td>Male</td>
<td>United-States</td>
<td>poor</td>
</tr>
<tr>
<td>4</td>
<td>Private</td>
<td>Bachelors</td>
<td>13</td>
<td>Married-civ-spouse</td>
<td>Prof-specialty</td>
<td>Wife</td>
<td>Black</td>
<td>Female</td>
<td>Cuba</td>
<td>poor</td>
</tr>
<tr>
<td>5</td>
<td>Private</td>
<td>Masters</td>
<td>14</td>
<td>Married-civ-spouse</td>
<td>Exec-</td>
<td>Wife</td>
<td>White</td>
<td>Female</td>
<td>United-States</td>
<td>poor</td>
</tr>
</tbody>
</table>
Filter Columns

Set Column Relevance Filter

Use the slider to select a subset of columns based on their relevance. If in doubt, do not change.

Guide

Set Column Relevance Filter

By default, all columns will be used to train the model that creates the prediction. However, not all columns contribute with the same importance or relevance to the final prediction. In some cases, columns are not informative or contain spurious information. To help you decide, the overall column relevance towards the final prediction is measured.

- **Column Relevance** is an overall metric summarizing the metrics belows. Use the slider to select the input features based on their Overall Column Relevance.

The additional metrics calculated automatically and used to determine Overall Column Relevance include:

- **ID/Noise Test** measures how likely the column is a representation used to identify each row in your table. Row identifiers are uninformative for your model and should be removed.
- **Constant Value Test** measures how often the column contains the exact same value. Columns with just a constant value also carry no information. You should avoid using them.
- **Missing Value Test** measures the
Select Models

Choose one or more machine learning models to train for your prediction task.

Simple models
- Naive Bayes
- Decision Tree
- Logistic Regression

Complex models
- Support Vector Machine
- Random Forest
- Generalized Linear Models
- Gradient Boosted Trees
- Deep Learning

Fine-tune Model Parameters

By default (unchecked), all parameters for the selected models and for feature engineering are automatically fully optimized. However, by selecting this option, you can fine-tune the model parameters to achieve better performance. This feature allows you to adjust the parameters of the models according to your specific requirements and data characteristics.

Guide

Select Models

Choose which models you want to train. The available models have different levels of complexity. Less complex models are simpler to interpret and understand and generally faster to train and more efficient to use in production. In contrast, more complex models are capable of solving more complicated problems at possibly a finer level of detail but possibly at the cost of longer training times and less efficient usage in production. If the time is not an issue and you simply want to see the best performing model, use all of the proposed models, especially the ones with higher complexity. You can later compare model performances as well as runtime to choose the model that best solves your task. If a convenient solution is what you are aiming for, enabling only the simpler models will save you training time and will allow for a more efficient execution.

If a convenient solution is what you are aiming for, enabling only the simpler models will save you both time to create and be more efficient in executing.

Levels of Complexity

- Simple models

  - Naive Bayes is a simple probabilistic classifier based on the Bayes' Theorem.
Execution Settings

Please select the desired distributed environments for the execution of the workflow.

Available options:

- Local execution
- Use Spark cluster if possible
- Use Apache Spark MLib
- Use other cluster environment

Guide

Execution Settings

By default, your model process will be run on your KNIME Server. There are different options to run the training of the selected models in distributed environments.

The options will depend on what you have available in your system:

- **Local execution**: all parts of the workflow will run on your KNIME Server.
- **Use Spark if possible**: parts of this workflow can be run in parallel on your Spark cluster, e.g., the H2O Sparkling Water integration can be used to train H2O models. The parts which cannot be run in parallel will run on your KNIME Server. If selected, and your cluster is available, training time might be reduced.
- **Use Apache Spark MLib**: several model learning and optimization algorithms can be executed on Spark using the Apache Spark MLib. If selected, and your cluster is available, training time might be reduced.
- **Use other cluster environment**: workflow execution is distributed on the other cluster environment.
Download Models

Here is a summary of information (performances) about the models trained based on your specifications. The first chart compares the accuracy and Area under the Curve of each model. The second chart compares the training times. The third chart compares the prediction time on a new record. The fourth chart shows the ROC (AUC). After the table to download the model parameters, a performance summary for each model is shown.

Compare Model Metric Performance

This bar chart visualizes different performance metrics to assess the quality of each model.

Main Performance Metrics
Download Models

Here is a summary of information (performances) about the models trained based on your specifications. The first chart compares the accuracy and Area under the Curve of each model. The second chart compares the training times. The third chart compares the prediction time on a new record. The fourth chart shows the ROC (AUC). After the table to download the model parameters, a performance summary for each model is shown.

Compare Model Metric Performance

This bar chart visualize different performance metrics to assess the quality of each model.

Main Performance Metrics
Compare Training and Prediction Times

The first bar chart compares the training times of all models. The second bar chart compares the prediction time for one single sample.

Training Time

- Naive Bayes
- Generalized Linear Model
- Decision Tree
- Gradient Boosted Tree
- Logistic Regression
- Deep Learning
- Random Forest

Prediction Time per Sample

- Generalized Linear Model
- Random Forest
- Logistic Regression
- Deep Learning
- Decision Tree
- Gradient Boosted Tree
- Naive Bayes

Advanced Assessment of Models

The advanced assessment of models sections shows four additional charts per model:

1. Performance Metrics Bar Charts
   - Recall (or True Positive Rate) (% of “rich” rows correctly classified)
   - Precision (or Positive Predicted Value) (% of predicted “rich” rows correctly classified)
   - Specificity (or True Negative Rate) (% of not “rich” rows correctly classified)
   - f-Maueau (Harmonic average between Recall and Precision)

2. Cumulative Gain Chart and Lift Chart
   This chart can display two different charts: the Cumulative Gain Chart and the Lift Chart. By default, the cumulative gain chart is displayed. This chart is drawing a curve that reflects how well the model is doing compared to a random classifier. You are selecting rows from the test ranked by the probability of class “rich”. On the x-axis you have the percentage of top ranked rows by the model that define the partition of rows you are considering. On the y-axis you measure the response as the percentage of “rich” rows over their total number in your partition of top ranked rows. If the model is bad, the curve will be close to the black line (random classifier), whereas the percentage of original “rich” rows is exactly equal the percentage of selected rows. The cumulative gain curve should be above the bisector line and the greater the area between the cumulative gain curve and the bisector line is, the better the model is.

3. Global Feature Importance Bar Chart
   This chart shows the global feature importance. A surrogate random forest model is trained overfitting the test set predicted classes. From such a model it is possible to measure how often each feature is useful to outcome a prediction. In the chart the six most important features are shown whereas only features of the original data set are considered. More information at this link.

4. Confusion Matrix Heatmap
   This chart shows a confusion matrix. A confusion matrix is summarizing all the predictions on the test set by considering how many instances fall in each cell according to prediction and ground truth. The heatmap is encoding with shades of blue the number of instances in each cell. A model with a good performance should have most of the dark blue cells on the diagonal from top-left corner to bottom right corner of the confusion matrix. More information at this link.
ROC Curve Plots

Plots the ROC curves, one for each model. The greater the area under a curve the better the model is. To plot this chart the following settings for the target Label were automatically defined:

- positive class: rich
- negative class: poor
Download Model

The following table summarizes the information in the charts. Please select the model you would like to download and use for predictions.

<table>
<thead>
<tr>
<th>Model</th>
<th>Accuracy (%)</th>
<th>Area Under Curve (%)</th>
<th>Prediction Time (ms)</th>
<th>Training Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>83.385</td>
<td>89.373</td>
<td>0.147</td>
<td>2.9</td>
</tr>
<tr>
<td>Gradient Boosted Trees</td>
<td>83.802</td>
<td>89.135</td>
<td>0.034</td>
<td>5</td>
</tr>
<tr>
<td>Logistic Regression</td>
<td>83.438</td>
<td>50</td>
<td>0.126</td>
<td>6.1</td>
</tr>
<tr>
<td>Deep Learning</td>
<td>83.258</td>
<td>88.173</td>
<td>0.123</td>
<td>11.7</td>
</tr>
<tr>
<td>Decision Tree</td>
<td>82.001</td>
<td>85.583</td>
<td>0.089</td>
<td>3.3</td>
</tr>
<tr>
<td>Generalized Linear Model</td>
<td>81.545</td>
<td>88.3</td>
<td>0.006</td>
<td>2</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>81.207</td>
<td>88.355</td>
<td>0.049</td>
<td>2.5</td>
</tr>
</tbody>
</table>

Showing 1 to 7 of 7 entries
Advanced Assessment of Models

Each row represents a series of additional information about each trained model.

- target feature: Label
- positive class: rich

### Decision Tree

Figure 1: Performance Metrics
- Recall (TPR)
- Specificity (TNR)
- Precision (PPV)

Figure 2: Cumulative Gain Chart

Figure 3: Global Feature Importance
- Education-Num
- Occupation
- Marital Status
- Country

Figure 4: Confusion Matrix Heatmap
- Actual Labels
- Predicted Labels

### Gradient Boosted Trees

Figure 5: Performance Metrics
- Recall (TPR)
- Specificity (TNR)
- Precision (PPV)

Figure 6: Cumulative Gain Chart

Figure 7: Global Feature Importance
- Education-Num
- Occupation
- Marital Status
- Country

Figure 8: Confusion Matrix Heatmap
- Actual Labels
- Predicted Labels
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.