DEUTSCHE TELEKOM AG
INDIVIDUAL SOLUTIONS & PRODUCTS
OPTIMIZED PREDICTIVE PLANNING WITH KNIME

Michael Schaarschmidt
Christian Gräber
March 2019
OPTIMIZED PREDICTIVE PLANNING WITH KNIME
FROM BUSINESS PROBLEM TO MODELING AND IMPLEMENTATION

business problem  implementation
OPTIMIZED PREDICTIVE PLANNING WITH KNIME
FROM BUSINESS PROBLEM TO MODELING AND IMPLEMENTATION

- Business problem
- Decision elements
- Underlying data
- Modelling
- Implementation
OPTIMIZED PREDICTIVE PLANNING WITH KNIME
FROM BUSINESS PROBLEM TO MODELING AND IMPLEMENTATION
BUSINESS PROBLEM

- Planning section
- Resource costs
- Order backlog
- Sales funnel
- Unknown business
Manuell planning time is too high in relation to the budget.
Consideration of the planning part with the highest resource requirements and lowest validity.
OPTIMIZED PREDICTIVE PLANNING WITH KNIME
FROM BUSINESS PROBLEM TO MODELING AND IMPLEMENTATION
DECISION ELEMENTS

- Regression Modell
- Cluster & Similarities
- Probability of orders
OPTIMIZED PREDICTIVE PLANNING WITH KNIME
FROM BUSINESS PROBLEM TO MODELING AND IMPLEMENTATION
EXAMPLES OF UNDERLYING DATA

Customer Dimension
- Sales Region
- Letter of Intent True/False

System Information
- Duration per Stage
- Number of offer versions

Time Dimension
- Quarter of the planned project start
- Condition per Stage

Financial Information
- project volume
- term of contract

Technology & Portfolio
- technology portfolio
- ITIL Type
OPTIMIZED PREDICTIVE PLANNING WITH KNIME
FROM BUSINESS PROBLEM TO MODELING AND IMPLEMENTATION

- Business problem
- Decision elements
- Underlying data
- Modelling
- Implementation
EVOLUTION OF THE MODEL FROM SIMPLE TO COMPLEX

1. Random Forest Learner

2. Python & Keras Network Learner
# Evolution of the Model from Simple to Complex

<table>
<thead>
<tr>
<th>1 Random Forest Learner</th>
<th>2 Python &amp; Keras Network Learner</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Easy to implement</strong></td>
<td></td>
</tr>
<tr>
<td>+ Can handle categorical values</td>
<td>+ Can recognize even complex relationships</td>
</tr>
<tr>
<td>+ No special data preparation required</td>
<td>+ Stable result even after retrain</td>
</tr>
<tr>
<td>+ Successful training even with smaller data sets</td>
<td></td>
</tr>
<tr>
<td><strong>Retrain creates a new model each time</strong></td>
<td><strong>High resource requirements for training</strong></td>
</tr>
<tr>
<td>- Small changes in the training data set can have a big impact on the model.</td>
<td>- Special data preparation required</td>
</tr>
<tr>
<td></td>
<td>- Scaling to range from -1 to 1 required</td>
</tr>
<tr>
<td></td>
<td>- Requires sufficient data for initial training (approx. 1,000 data rows per feature)</td>
</tr>
</tbody>
</table>
OPTIMIZED PREDICTIVE PLANNING WITH KNIME
FROM BUSINESS PROBLEM TO MODELING AND IMPLEMENTATION

business problem -> decision elements -> underlying data -> modelling -> implementation
INTEGRATION INTO DATA WAREHOUSE

- Execution times of individual Extract Load Transform (ETL) processes are defined in metadata management.
- Each transaction is traceable system-wide.
- Workflows of the individual processes read metadata, metadata controls workflows of individual processes.

<table>
<thead>
<tr>
<th>ID</th>
<th>name</th>
<th>starttime</th>
<th>last sucessful extraction time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>raw_data_workflow</td>
<td>2019-02-25 00:00:00.000000000</td>
<td>2019-02-25 00:10:00.000000000</td>
</tr>
<tr>
<td>2</td>
<td>historical_workflow</td>
<td>2019-02-25 00:10:01.000000000</td>
<td>2019-02-25 00:20:00.000000000</td>
</tr>
<tr>
<td>3</td>
<td>KNIME_workflow</td>
<td>2019-02-25 00:25:00.000000000</td>
<td>2019-02-25 00:50:00.000000000</td>
</tr>
<tr>
<td>4</td>
<td>raw_data_workflow</td>
<td>2019-02-25 01:00:00.000000000</td>
<td>2019-02-25 01:10:00.000000000</td>
</tr>
</tbody>
</table>
### ADDITIONAL FRAMEWORKS SUPPORTING DEEP LEARNING

<table>
<thead>
<tr>
<th>Framework</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Python & Anaconda** | - Installation of Python & custom Anaconda environment  
- Defining the Uniform Configuration for the KNIME Server Executor |
| **Tensorflow** | - Underlying framework  
- Currently as CPU version, GPU planned 2nd half of 2019 |
| **Keras** | - Implemented with Tensorflow backend |
OPTIMIZED PREDICTIVE PLANNING WITH KNIME

→ Increase of accuracy of forecast and decrease of resources needed

Available options: Predictor, KNIME model factory, neural network, testing phase, AI approach.
Thank you for your attention.