Open for Innovation®
KNIME

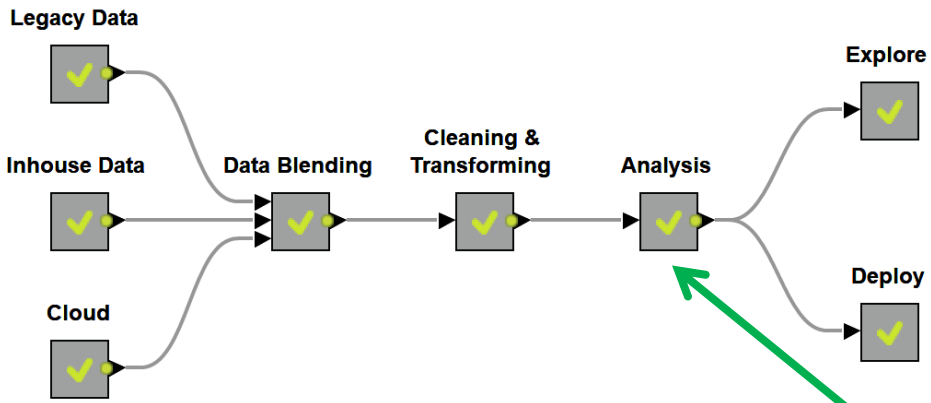
Guided Automation

Christian Dietz, Paolo Tamagnini

20 March 2019

#KNIMESummit2019

Automating What?

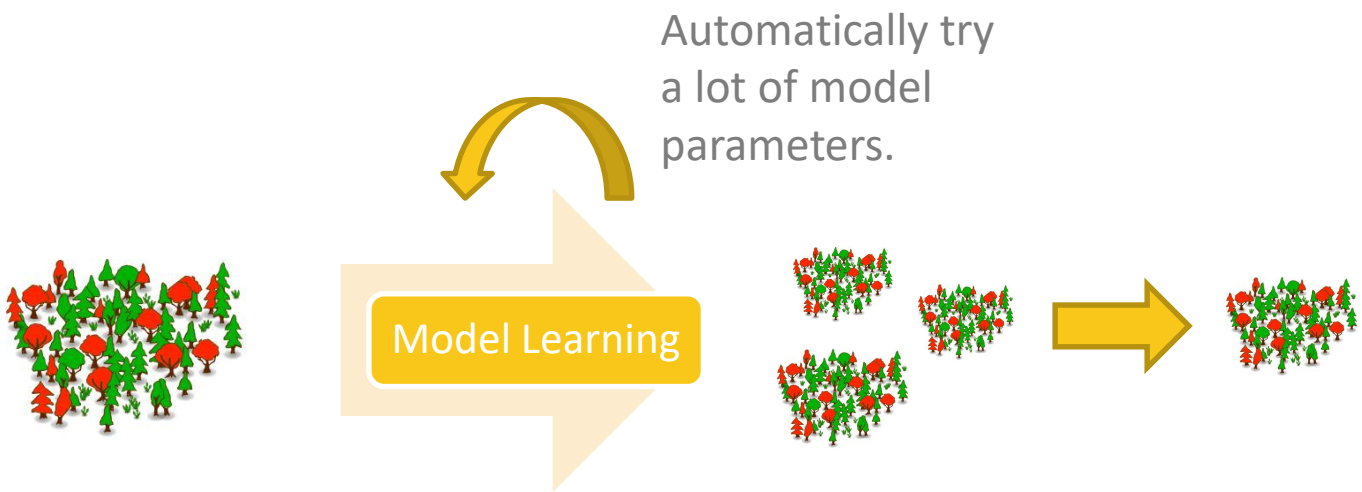


Automating Analytics:

- Parameter Optimization
- Model Selection
- Ensemble Construction

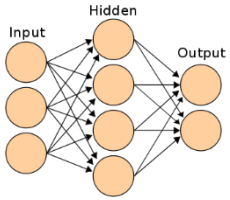
Automating What?

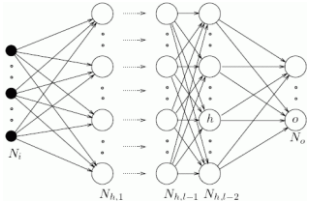
- Model Parameter Optimization

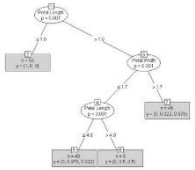


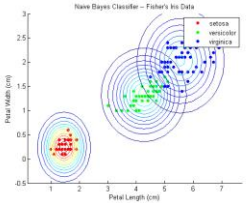
Automating What?

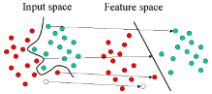
- Model Selection
  - Try many models, but in an automated way.





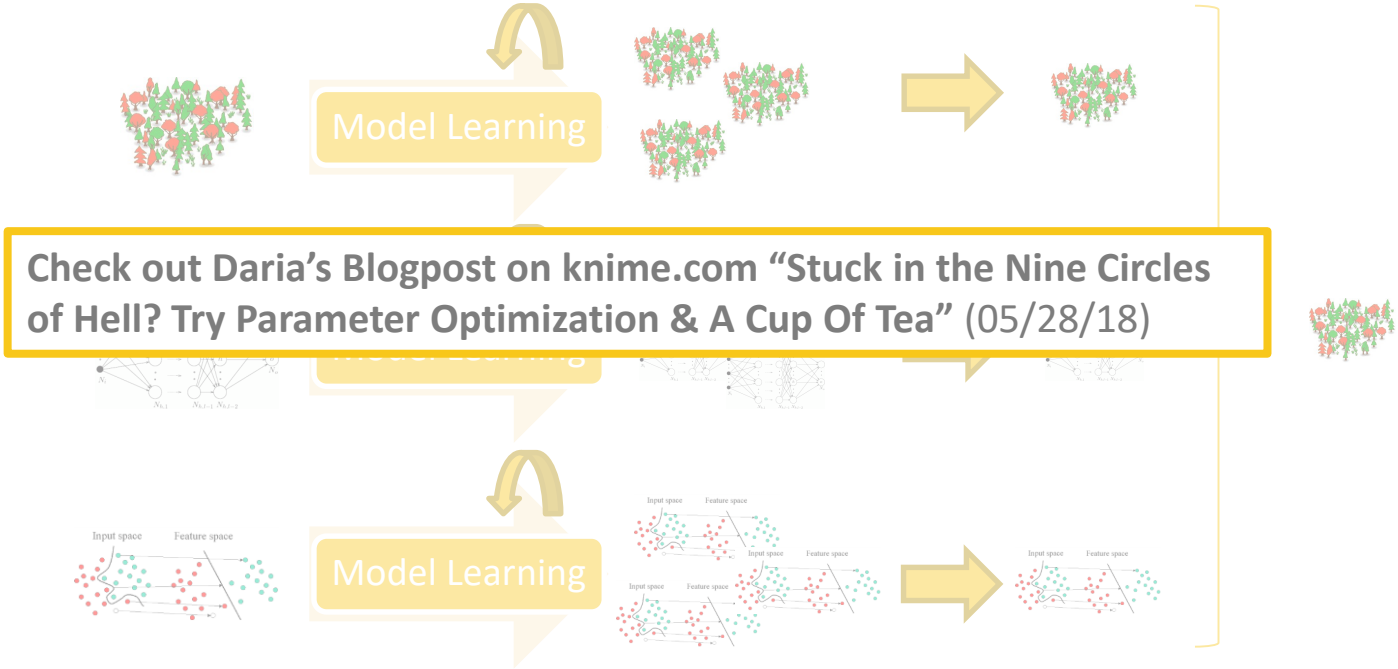






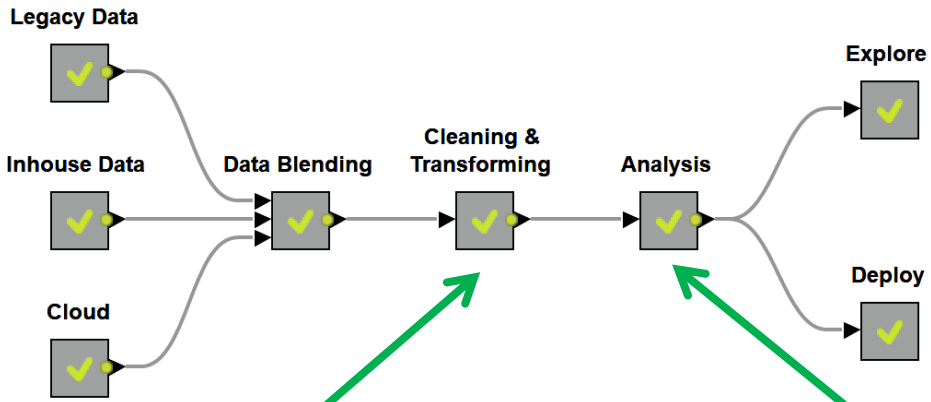
Automating What?

- 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

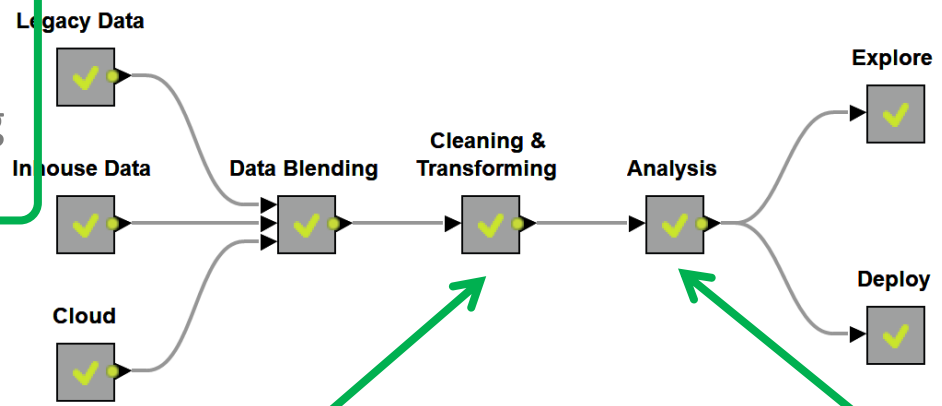
Automating What?

- 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

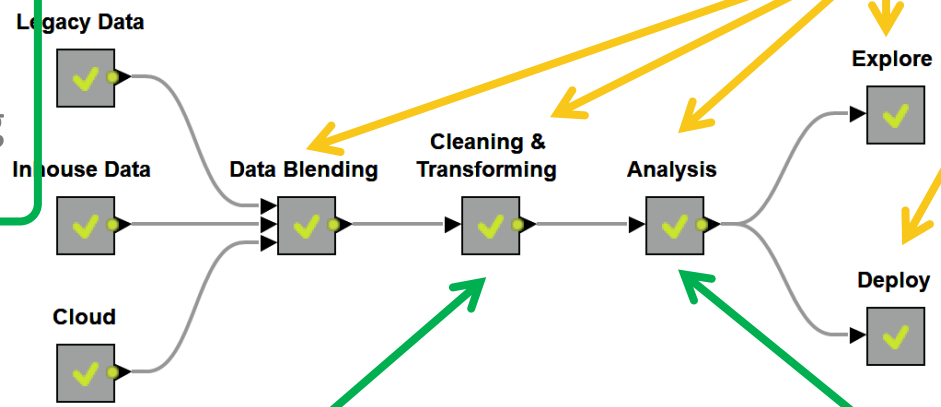
Automating Everything?

Human Input Needed!

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

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?

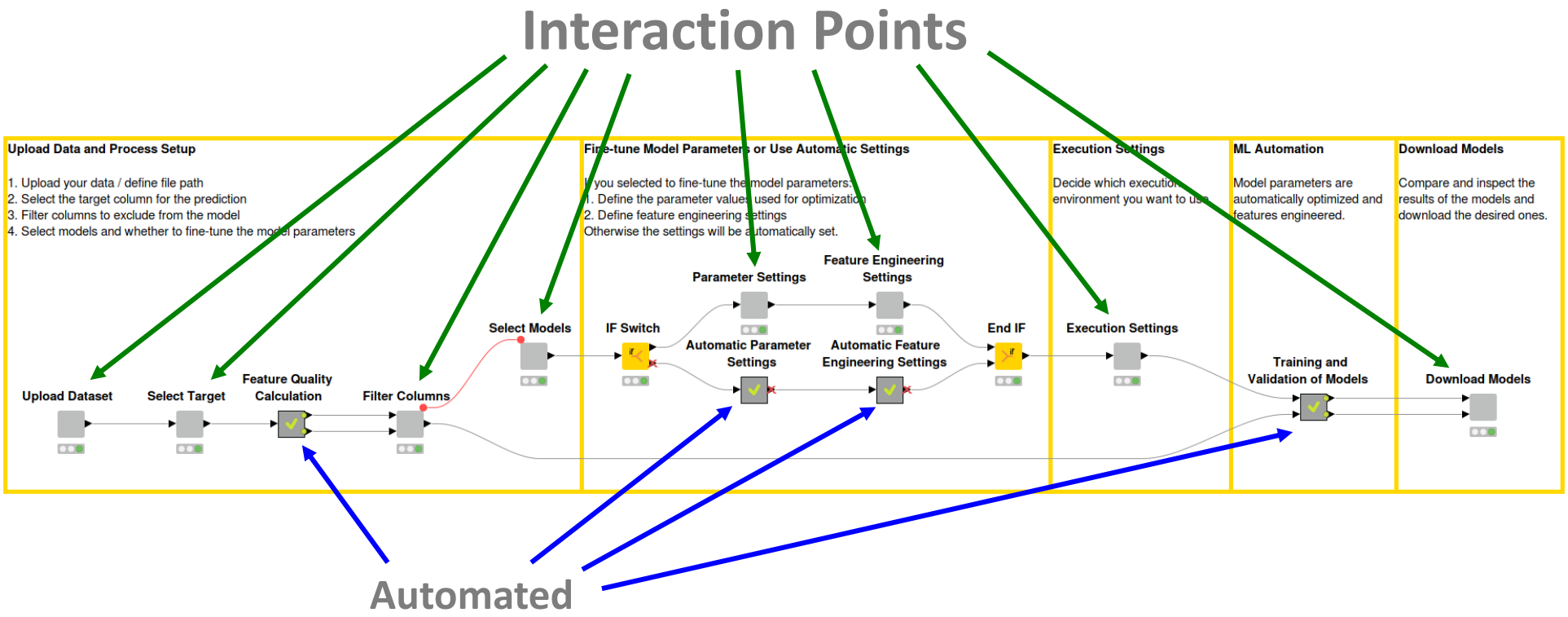
- Try: (Brute Force)
  - Use compute power to try variations
- Copy: (Transfer Learning)
  - Use similar problems as starting point
- Learn: (Bayesian Optimization)
  - Use Machine Learning to predict/learn parameter space
- Interact: (Guided Automation)
  - Augment Automation with Human Intelligence

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

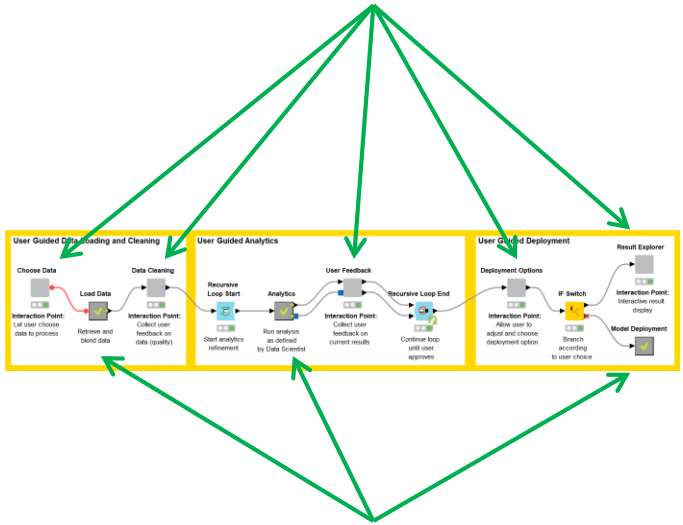
(How much interaction? That depends…)

Building a Guided Automation Workflow



KNIME's Guided Automation: Automation + Interaction

Interaction Points

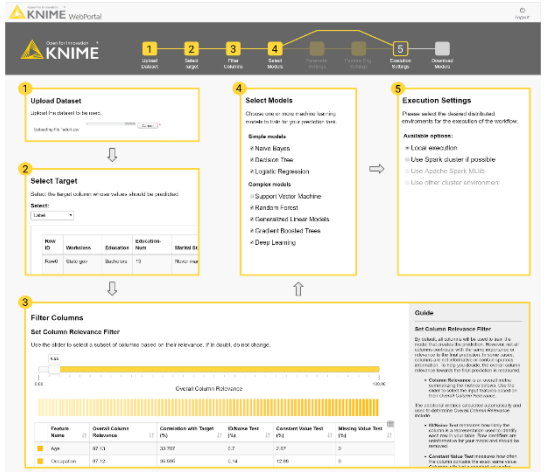


Automated

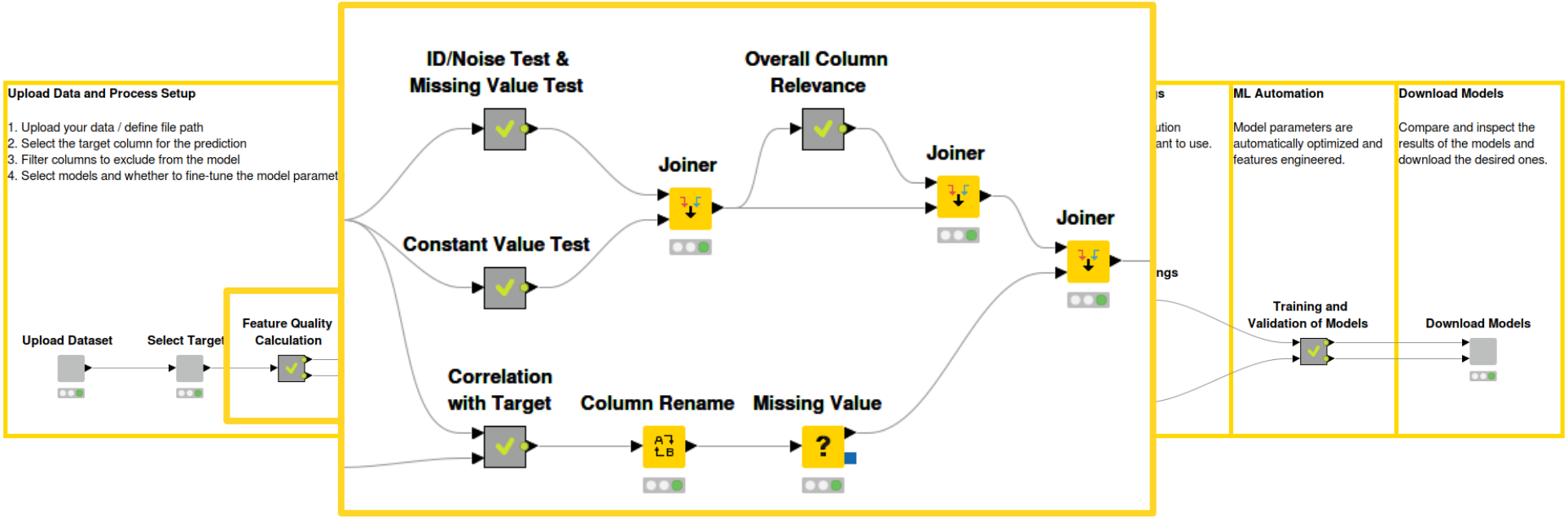
(prepackaged, updated, or created by in-house experts)

KNIME Server

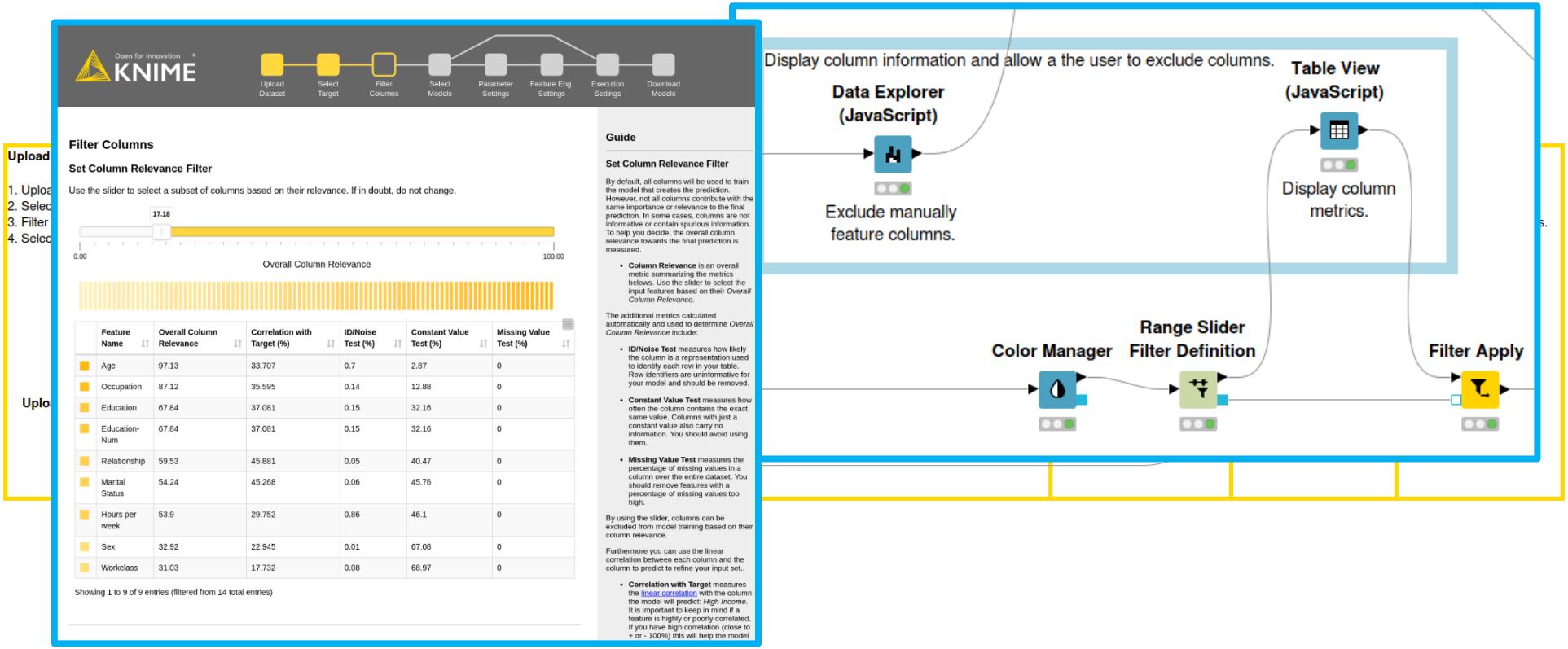
www.myserver.ch



Guided Automation: Automation + Interaction

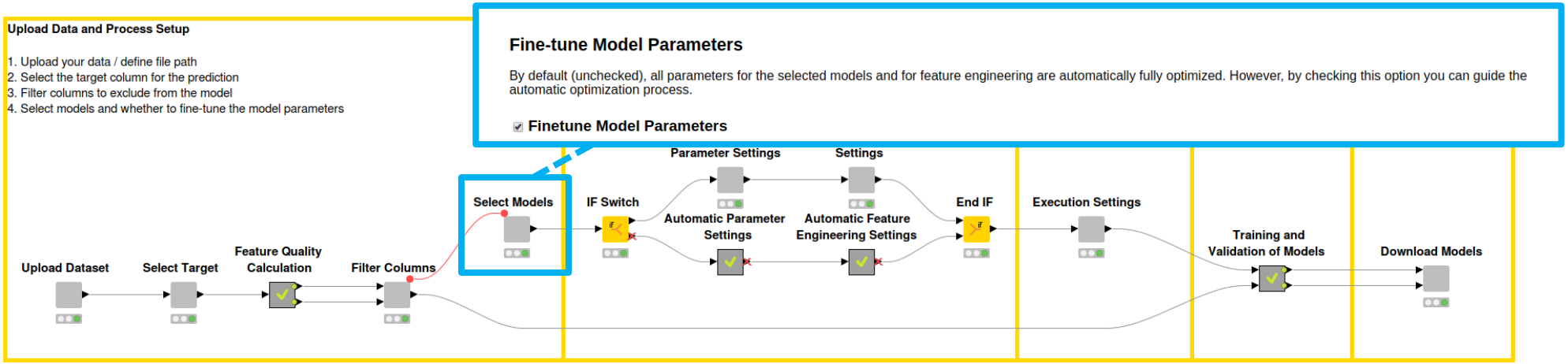


Guided Automation: Automation + Interaction

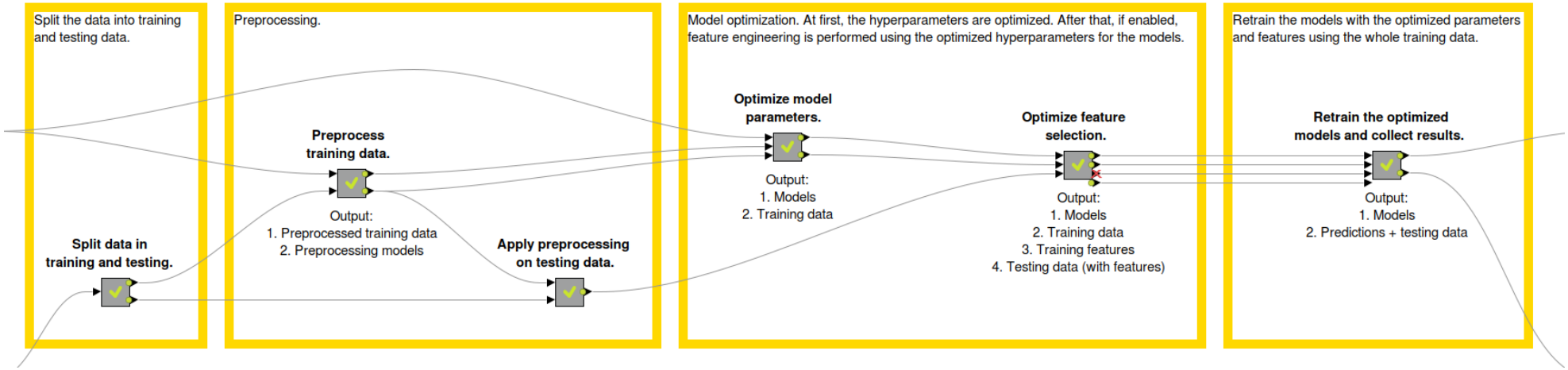


Open for Innovation ® KNIME

Guided Automation: Automation + Interaction



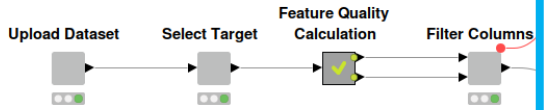
Guided Automation: Automation + Interaction

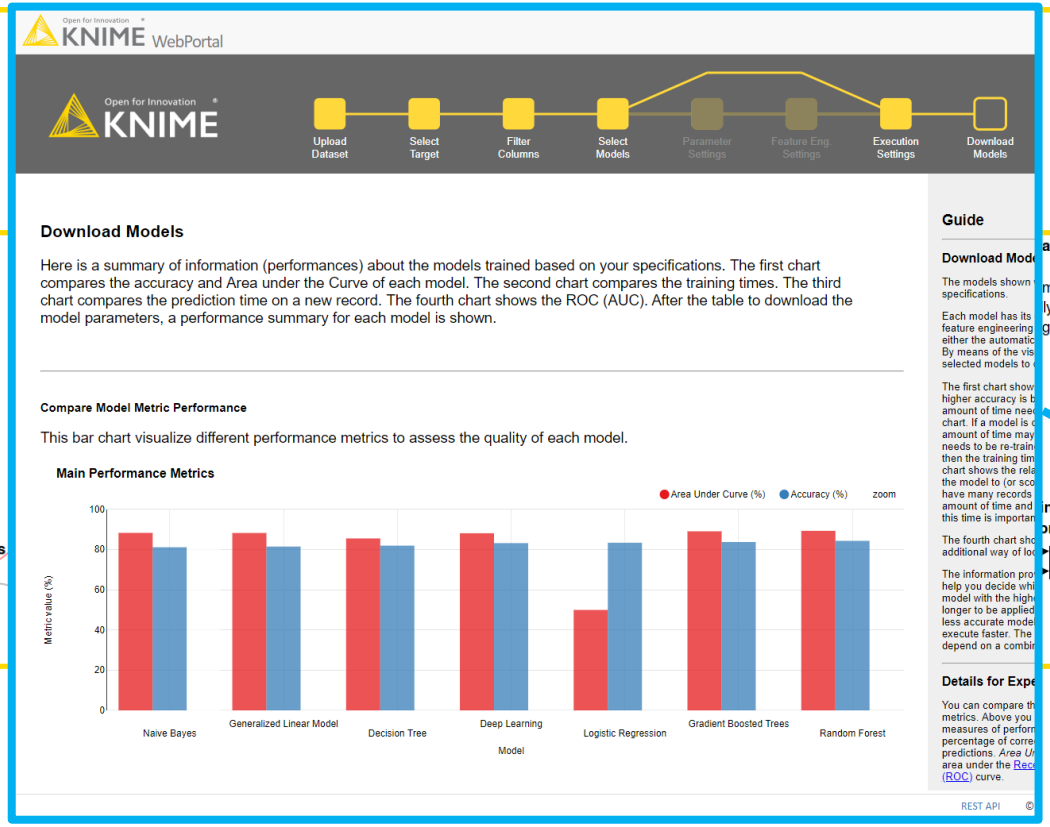


Guided Automation: Automation + Interaction

Upload Data and Process Setup

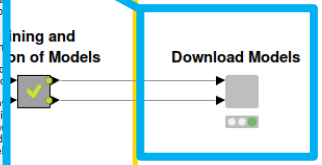
1. Upload your data / define file path
2. Select the target column for the prediction
3. Filter columns to exclude from the model
4. Select models and whether to fine-tune the model parameters





Download Models

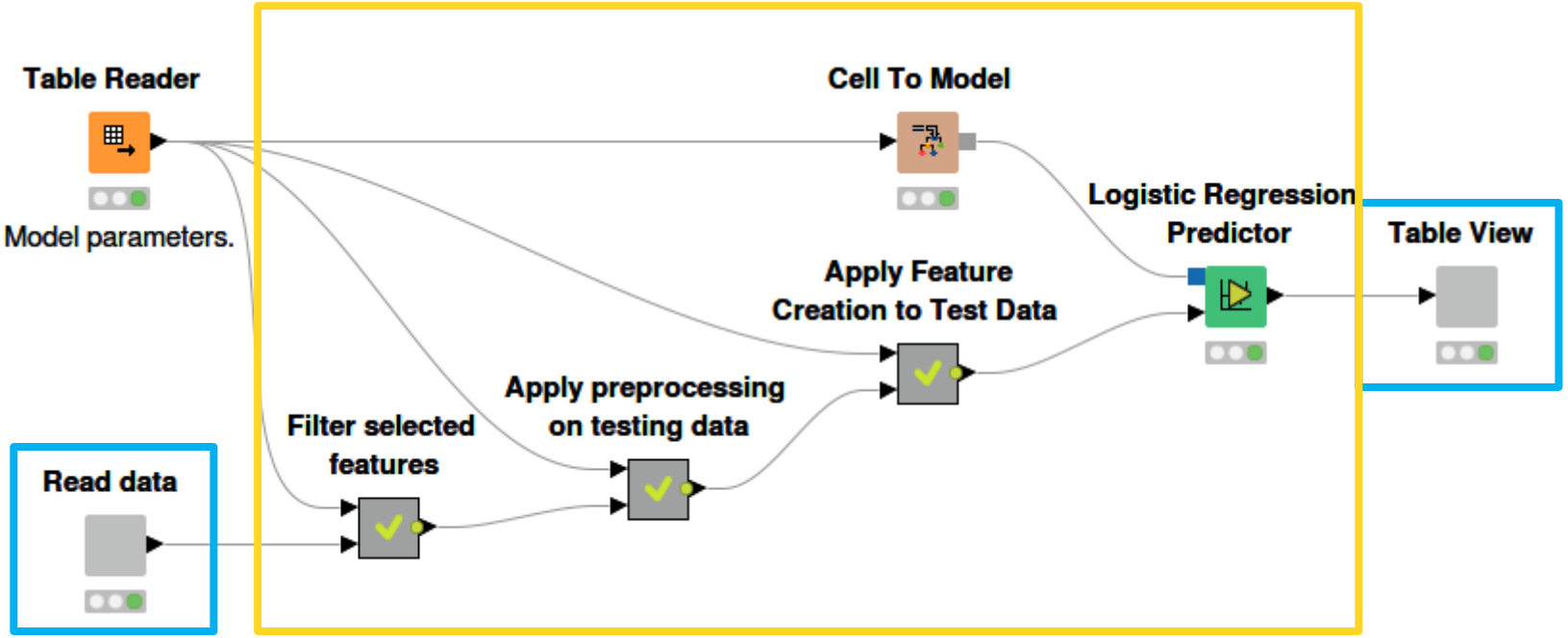
Compare and inspect the results of the models and download the desired ones.



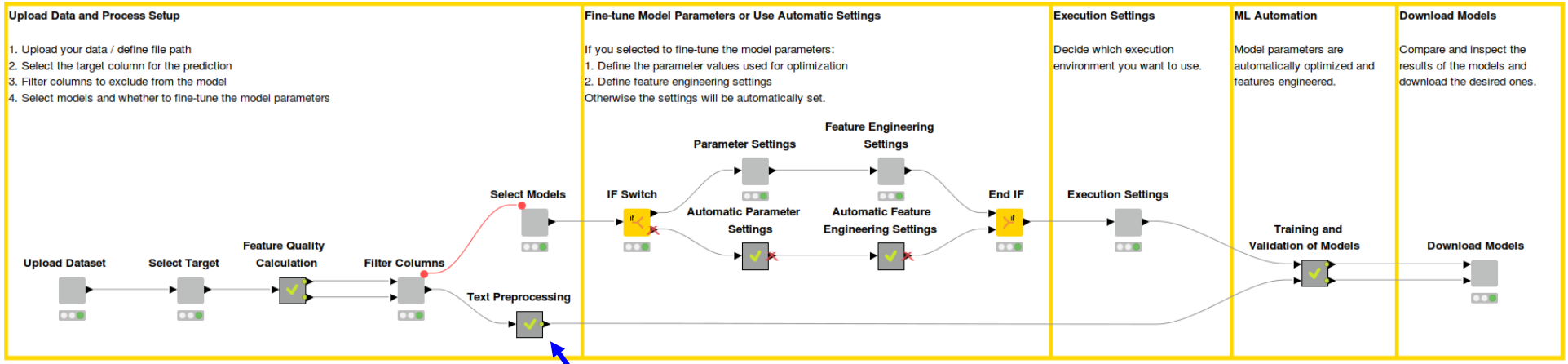
Guided Automation: Automation + Interaction

- Live Demo

Scoring Workflow



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.

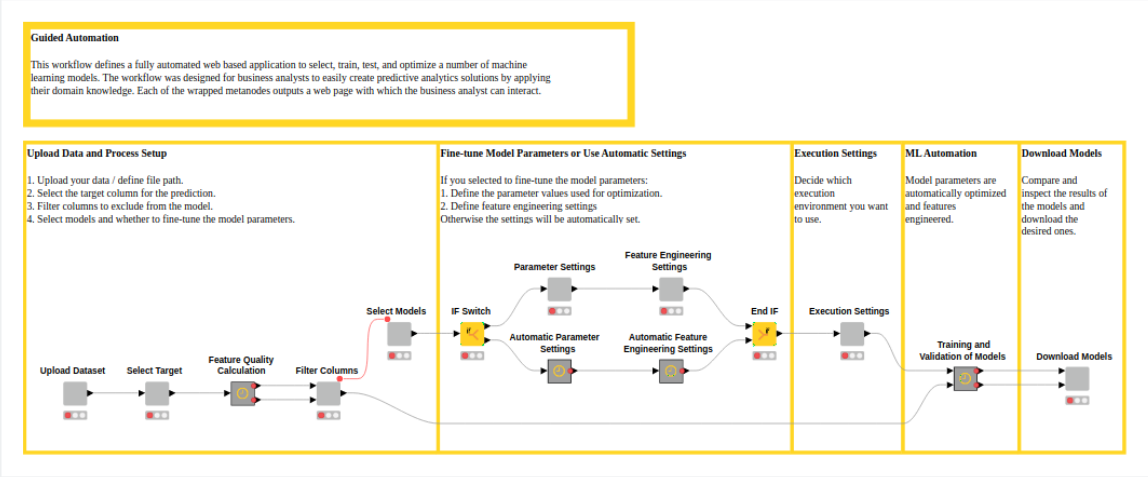
Workflow on the Hub

KNIME

Search workflows, nodes and more...

New to KNIME? More KNIME

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.

Simon Schmid

KNIME Team Member

Software developer and data scientist at KNIME.

Leader · Silver Anniversary · Thoughtful · 12 more

hosted by

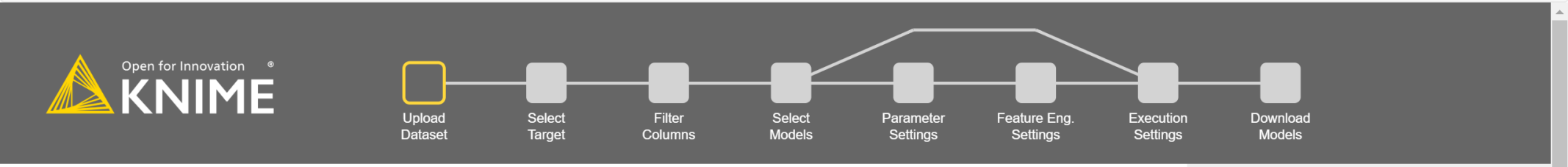
KNIME

Download workflow

By downloading the workflow, you agree to our terms and conditions.

License CC-BY-4.0

Open for Innovation ® KNIME



Upload Dataset

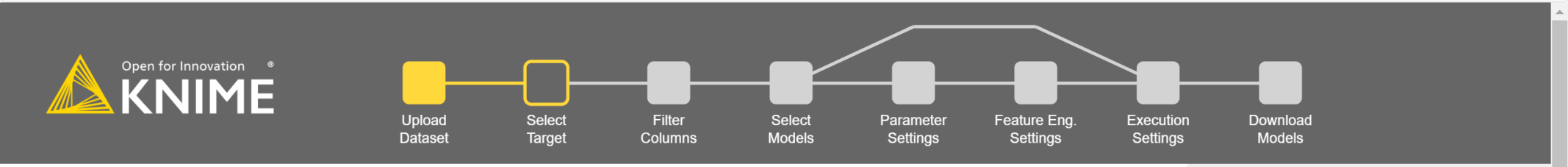
Upload the dataset to be used.

Uploading file "adult.csv" Cancel *

Guide

Upload Dataset

Upload the dataset to train the model. The dataset must be a representative sample of the past data history for the prediction problem at hand. Your file should be a KNIME table or a CSV file. The data will be uploaded onto the server for further processing.



Select Target

Select the target column whose values should be predicted.

Select:

Label

Row ID	Workclass	Education	Education-Num	Marital Status	Occupation	Relationship	Race	Sex	Country	Label
Row0	State-gov	Bachelors	13	Never-married	Adm-clerical	Not-in-family	White	Male	United-States	poor
Row1	Self-emp-not-inc	Bachelors	13	Married-civ-spouse	Exec-managerial	Husband	White	Male	United-States	poor
Row2	Private	HS-grad	9	Divorced	Handlers-cleaners	Not-in-family	White	Male	United-States	poor
Row3	Private	11th	7	Married-civ-spouse	Handlers-cleaners	Husband	Black	Male	United-States	poor
Row4	Private	Bachelors	13	Married-civ-spouse	Prof-specialty	Wife	Black	Female	Cuba	poor
Row5	Private	Masters	14	Married-civ-spouse	Exec-	Wife	White	Female	United-	poor

Guide

Select Target

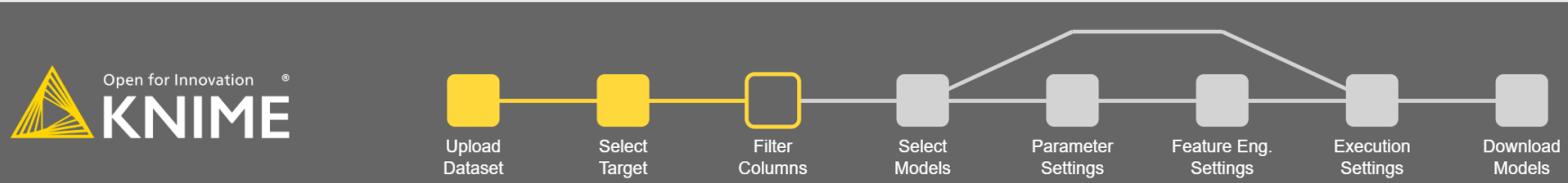
To train a model, a target column must be selected. In case of doubt, explore the preview (top 10 rows) on the left of all possible target columns for this prediction problem.

Details for Experts

The possible columns can be of different types. Different Machine Learning methods can be used with different types of targets. Based on the type of the selected target column, only valid methods for the prediction task will be surfaced.

List of prediction tasks:

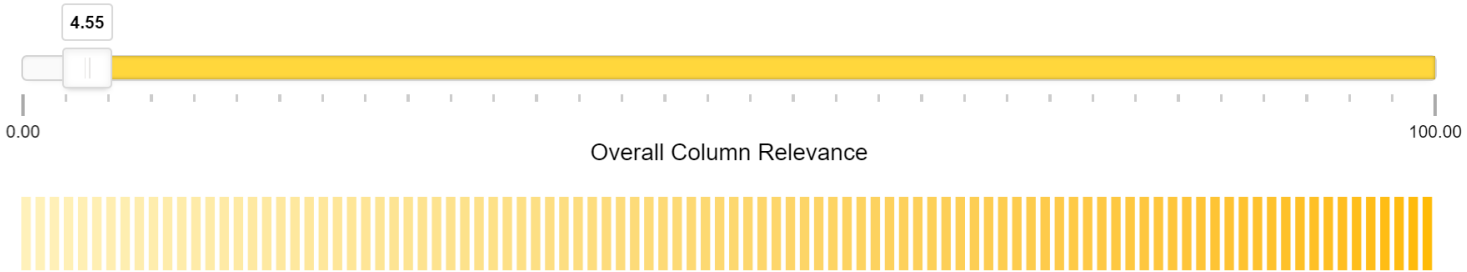
- If the target column is a categorical feature with 2 possible values, then it is a *Binary Classification* task.
- If the target column is a categorical feature with more than 2 possible values, then it is a *Multiclass Classification* task.
- If the target column is a numerical feature with only 2 different numbers, then it is a *Binary Classification* task.
- If the target column is a numerical feature with more than 2 and not more than 20 different numbers, then it is a *Multiclass*



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.



Feature Name	Overall Column Relevance	Correlation with Target (%)	ID/Noise Test (%)	Constant Value Test (%)	Missing Value Test (%)
Age	97.13	33.707	0.7	2.87	0
Occupation	87.12	35.595	0.14	12.88	0
Education	67.84	37.081	0.15	32.16	0
Education-Num	67.84	37.081	0.15	32.16	0

Guide

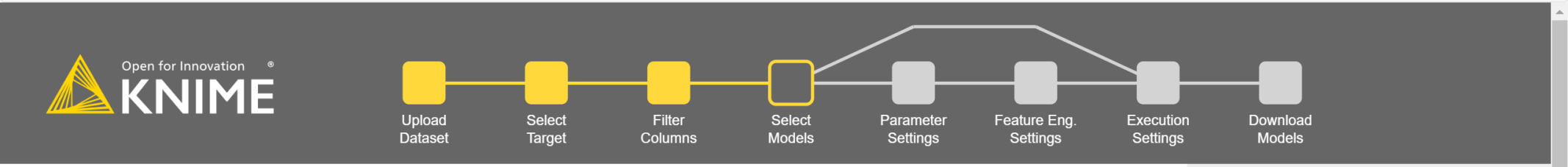
Set Column Relevance Filter

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

- [x] Naive Bayes
- [x] Decision Tree
- [x] Logistic Regression

Complex models

- [ ] Support Vector Machine
- [x] Random Forest
- [x] Generalized Linear Models
- [x] Gradient Boosted Trees
- [x] 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

Guide

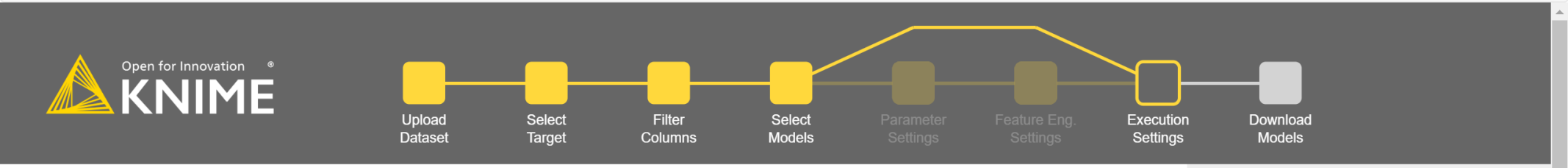
Select Models

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 enviroments for the execution of the workflow.

Available options:

- (●) Local execution
- ( ) Use Spark cluster if possible
- ( ) Use Apache Spark MLlib
- ( ) 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 MLlib**: several model learning and optimization algorithms can be executed on Spark using the Apache Spark MLlib. 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.

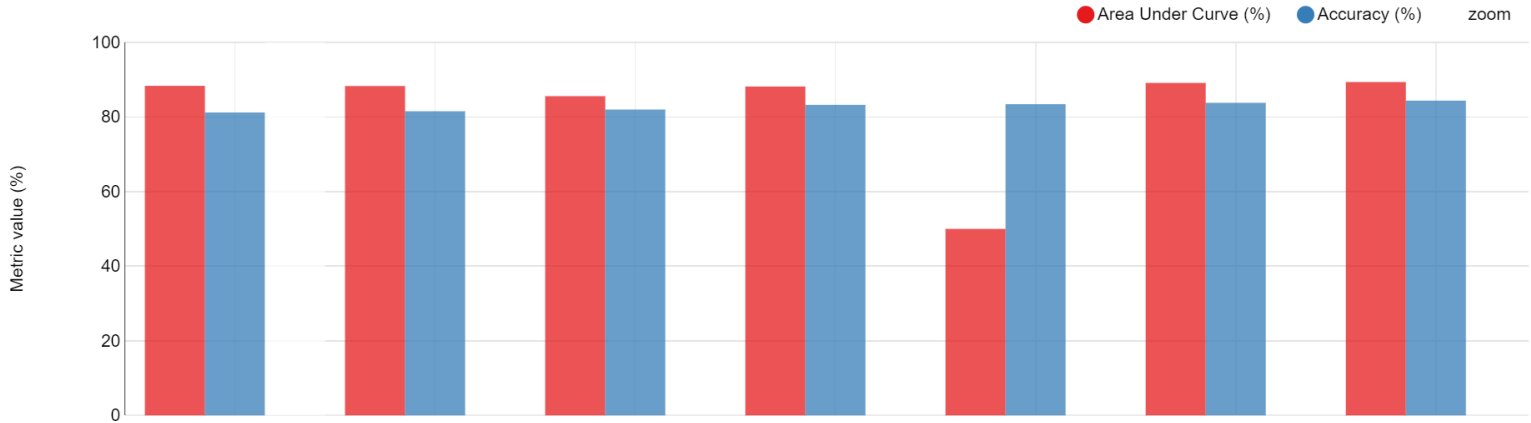
Download Models

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

Main Performance Metrics



Guide

Download Models

The models shown were trained based on your specifications.

Each model has its own hyperparameter optimization, feature engineering and feature selection based on either the automatic settings or the manual settings. By means of the visualizations below, compare the selected models to decide which model to deploy.

The first chart shows model accuracy and AUC. A higher accuracy is better than a lower accuracy. The amount of time needed to train a model is in the next chart. If a model is only trained occasionally then the amount of time may be irrelevant. If a new model needs to be re-trained more often (daily, hourly, etc.) then the training time may be important. The next chart shows the relative time it would take to apply the model to (or score on) a new record. When you have many records that need to be scored in a short amount of time and possibly at a frequent rate then this time is important.

The fourth chart shows the ROC curve. It provides an additional way of looking at model performance.

The information provided by the four charts should help you decide which model is most suitable. The model with the highest accuracy may take much longer to be applied to (or scored on) new data than less accurate models. These last ones however might execute faster. The “right” model for your situation will depend on a combination of all these factors.

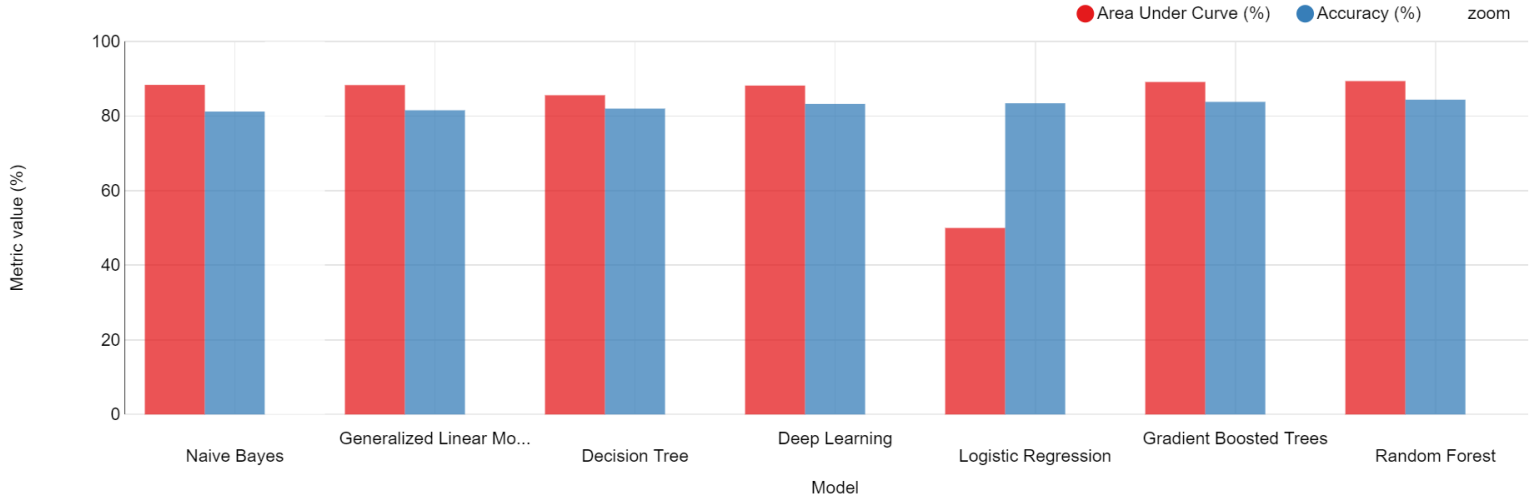
Download Models

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



Guide

Download Models

The models shown were trained based on your specifications.

Each model has its own hyperparameter optimization, feature engineering and feature selection based on either the automatic settings or the manual settings. By means of the visualizations below, compare the selected models to decide which model to deploy.

The first chart shows model accuracy and AUC. A higher accuracy is better than a lower accuracy. The amount of time needed to train a model is in the next chart. If a model is only trained occasionally then the amount of time may be irrelevant. If a new model needs to be re-trained more often (daily, hourly, etc.) then the training time may be important. The next chart shows the relative time it would take to apply the model to (or score on) a new record. When you have many records that need to be scored in a short amount of time and possibly at a frequent rate then this time is important.

The fourth chart shows the ROC curve. It provides an additional way of looking at model performance.

The information provided by the four charts should help you decide which model is most suitable. The model with the highest accuracy may take much longer to be applied to (or scored on) new data than less accurate models. These last ones however might execute faster. The “right” model for your situation will depend on a combination of all these factors.

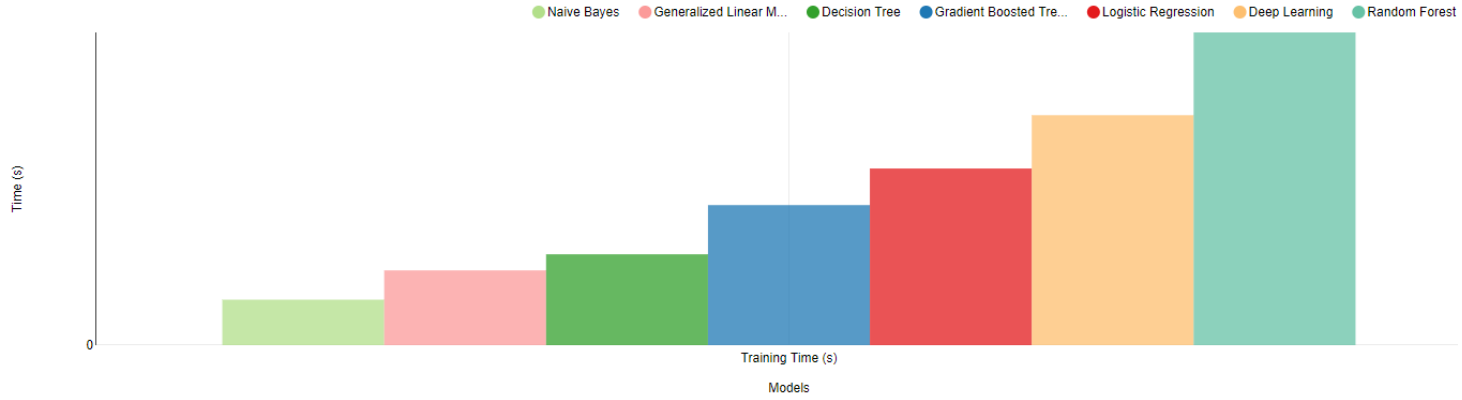
Details for Experts

You can compare the models now by different metrics. Above you can see a bar chart with two main measures of performance. *Accuracy* is the

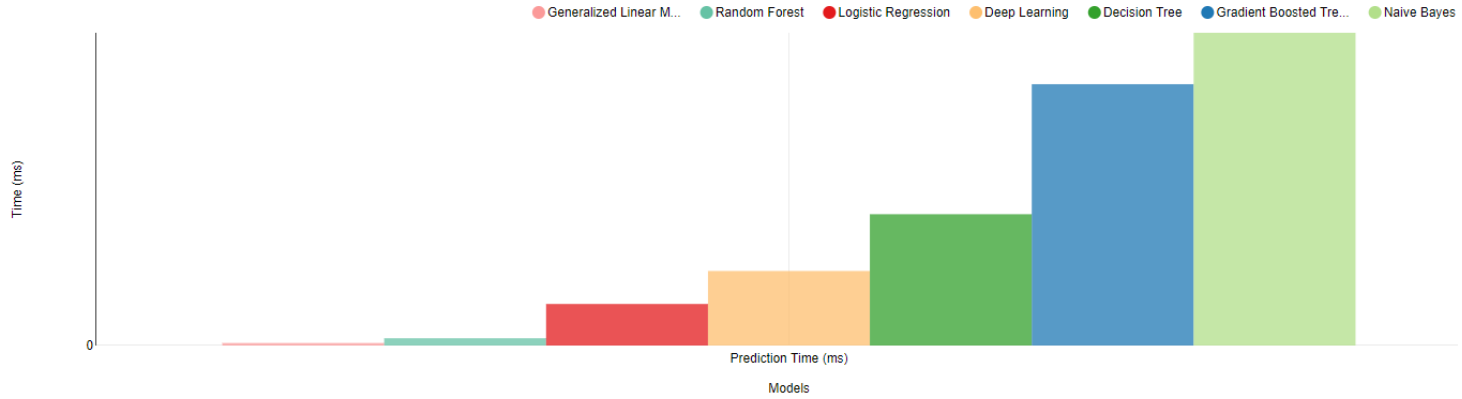
Compare Training and Prediction Times

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



Prediction Time per Sample



Advanced Assessment of Models

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

- **1. Performance Metrics Bar Charts**

  For this visualization we measured the following metrics:

  - *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-measure* (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), where 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.

  If you click on the top right corner of this chart, you will be able to visualize the relative lift chart as well. The lift on the y-axis measure the different between the cumulative gain chart curve and the bisector line.

- **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 whereby only features of the original data set are considered. More information at this link.

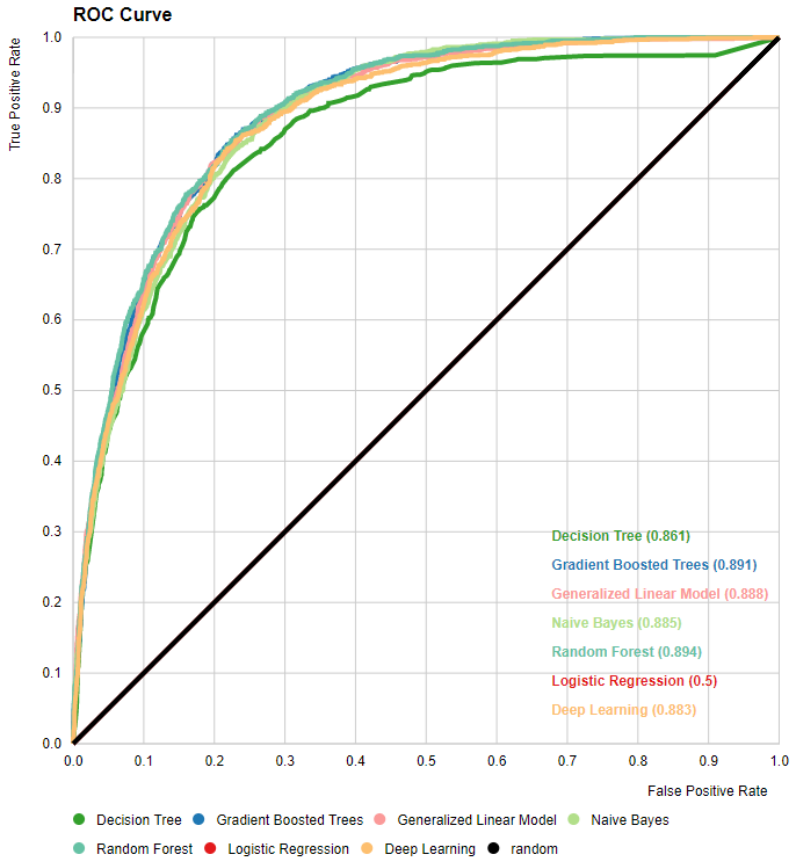
- **4. Confusion Matrix Heatmap**

  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.

Model	Accuracy (%)	Area Under Curve (%)	Prediction Time (ms)	Training Time (s)
Random Forest	84.385	89.373	0.147	2.9
Gradient Boosted Trees	83.802	89.135	0.034	5
Logistic Regression	83.438	50	0.126	6.1
Deep Learning	83.258	88.173	0.123	11.7
Decision Tree	82.001	85.583	0.089	3.3
Generalized Linear Model	81.545	88.3	0.006	2
Naive Bayes	81.207	88.355	0.049	2.5

Showing 1 to 7 of 7 entries

Decision Tree

Download Workflow

Gradient Boosted Trees

Download Workflow

Generalized Linear Model

Download Workflow

Naive Bayes

Download Workflow

Random Forest

Download Workflow

Logistic Regression

Download Workflow

Deep Learning

Download Workflow

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

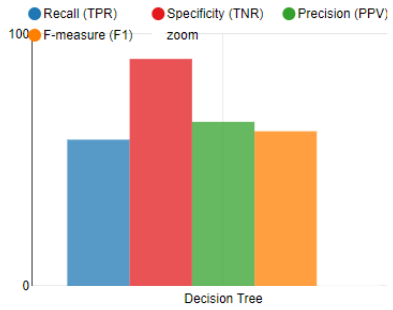


Figure 2: Cumulative Gain Chart

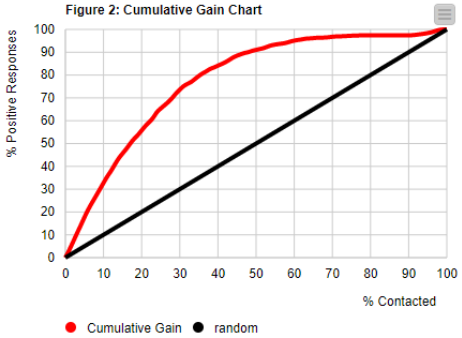


Figure 3: Global Feature Importance

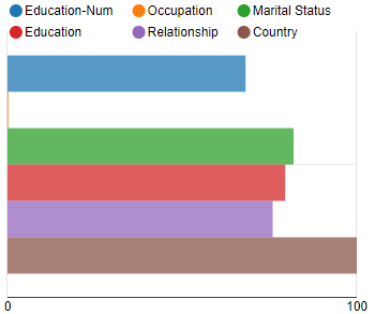
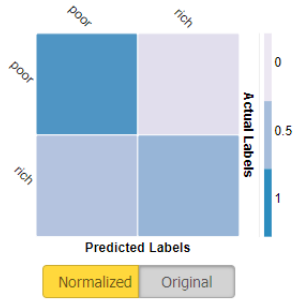


Figure 4: Confusion Matrix Heatmap



Gradient Boosted Trees

Figure 1: Performance Metrics

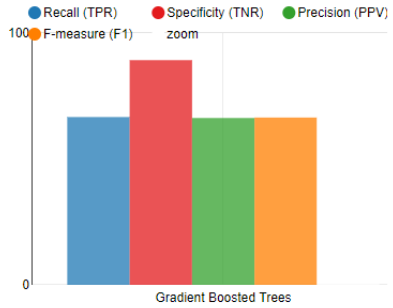


Figure 2: Cumulative Gain Chart

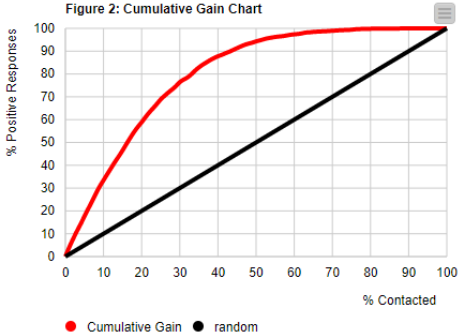


Figure 3: Global Feature Importance

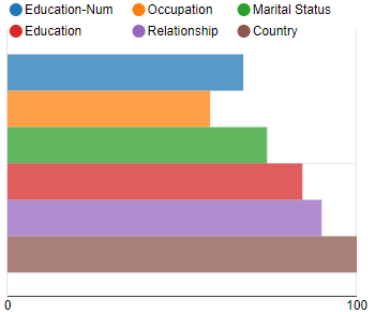


Figure 4: Confusion Matrix Heatmap