Practicing Data Science with Knime

DRAFT - subject to change

Sebastian Sauer

12/7/2020

Contents

Case Study: Titanic disaster **6**

Intro to Knime

What is Knime?

Knime Analytics Platform is a data science software with a graph-based GUI; no need for coding. Read more here.

Check out Wikipedia's site on Knime.

Download

Download the software (all major platforms) here.

Resources

There are plenty of resources got get your started; here's a curated (opinionated) selection:

- The Knime Hub
- Cheat Sheets, particularly Building a KNIME Workflow for Beginners
- Example Workflows
- Added functionality via Components
- Knime books
- Knime courses
- Knime White Papers
- The general Knime Learning Program
- Knime docs
- Knime forum
- Knime Youtube channel
- Knime self-paced courses
- Knime course at Coursera, freemium licence

Diving deeper into theory

- Have a glimpse at some of the machine learning algos implemented in Knime.
- Familiarise yourself with (at least) the core ideas of statistical learning, eg., via An Introduction to Statistical Learning; a great book, free access (watch out for upcoming second edition!)

Setup

- Download the software.
- Read the cheat sheet Building a KNIME Workflow for Beginners.
- Get to know to the Knime Workbench.
- Make sure your internet connection is stable; you may need to install Knime extensions, and we will download and upload a number of items (may include largish data).
- Depending on your system, you may need admin rights, in order to install Knime extensions. Please try to install a Knime extension upfront.

Let's get going!

First work in Knime

- Get to know to your Knime cockpit

First Workflow

- Implement this workflow, a workflow that visualizes sales data.
- The workflow is available in Knime under examples: `Examples > 02_ETL_Data_Manipulation > 00_Basic_Examples > 00_Visual_Analysis_of_Sales_Data`
- Note that if you would like to change the color scheme, there's a tick box you need to check; see under "General options" and tick "Use row colors".
- In addition, you might need to install the Knime Extension BIRT to see a static version of created images (right click and hit the magnifier icon to see the created images).
- Similarly, to save the created image to disk, add the node "Image Writer (Port)".
- Check out this workflow, where the aforementioned pieces have been implemented.

Didactic outline of this course

A number of *guided practices* (GP) and *exercises* (Ex) are provided in this course. In the first step, the instructor will present some materials alongside with concepts and some how-to (GP). This is followed by exercises (without instantaneous solutions) to be solved by the participants. Discussion in the plenary ensues.

(Big) data wrangling

Guided Practice 1: Data Wrangling

- Download this workflow: *Example Workflow for ETL Basics Operations*; alternatively you'll find it in the Knime examples: `Examples < 02_ETL_Data_Manipulation > 00_Basic_Examples > 02_ETL_Basics`.
- Follow the steps outlined by the instructor.
- Checkout the Knime help (`description` field in the Knime workbench) for each node.

NOTE: There appears to be a bug in the string-to-date conversion. Check out this improved (?) version, where the bug is solved.

Ex1: (Big) data wrangling

Let's explore the NYC Yellow cab taxi data set (source).

Ex1a: Largish data set 1 (500k rows)

The following workflow demonstrates typical data wrangling steps in "largish" data (some hundred MBs).

Use the June 2020 Yellow Cab data for modelling.

Objective: Get max tip proportion per hour

- Compute the maximal tip proportion (in relation to total fare) per hour of day. Refine to credit card payment only.
- Visualize this statistic.

GP1: Redo this workflow in Knime

- Follow the steps shown by the instructor.
- Import this workflow.
- Rebuild the workflow from scratch. Pick and choose the nodes from the knode repo in your Knime app. Be sure to configure the nodes accordingly. Note that while you can copy-pase nodes from the workflow provided, I suggest that you build the workflow yourself (so that you learn more).

Redo this workflow in Excel Try to convert the workflow to Excel (particularly useful for Excel aficionados).

Redo this workflow in R See here for a in-principle-solution.

Ex1b: Largish data set 1 (1500k rows)

Here come the same operations, but the data set is large (~500 MB, 1500k rows).

- Download the 2020 January Yellow Cab taxi datta set.
- Redo the analysis in Knime, Excel, and R. See here for a type of solution in the R way.

Data Vizualization

GP1: Compose a scatterplot

- Find the workflow in your Knime explorer: Examples > Data Viz > Java Script > 12 Bivariate
- Alternative, download from Knime hub
- Redo the steps outlined by the instructor.

Ex1: Prices of diamonds

Data set

First, download this data set. See here for the data dictionary.

Objective

Visualize the distribution of the price, grouped by cut. Add the mean and/or the median to the picture for each subgroup.

Solution in Knime without saving to disk

Check out this solution.

Solution in Knime with saving to disk

Check out this solution.

Similarly, here's a solution *save to disk* for the example "Visual Analysis of the Sales Data"

Solution in R

Check out this solution.

Ex1: Movie budgets

Data set

First, download this data set. See here for the data dictionary.

Objective

Visualize the association of movie budget and movie rating, separed for each genre.

Solution in Knime

Please provide :-)

Solution in R

See this case study.

Case Study: Titanic disaster

Objective

Try to predict who ~~will~~ has drowned (died) and who survived. This is a famous competition for practicing data science, as can be seen in this Kaggle competition.

Data set

Note that the data consists of two parts: The *train* data set and the *test* data set.

You can access the data from here. Get the train data set here; and the test data set here.

The data is also available from the Kaggle competition page.

Kepp in mind: Our objective is to predict `Survived`.

Kaggle competition (optional)

Engage in a Kaggle competition for this case study (if you want). Sign up for Kaggle, log in and enrol to the Titanic competition.

The beauty is that your test competition will get scored, so you'll get some kind of objective feedback on the quality of your predictions.

Kaggle scores

Feel freee to check out my Kaggle scores in the Titanic competition. Don't expect a miracle though; I've just been playing . . . .

Logistic regression

GP1

- Download this workflow or find it in Knime Explorer (Examples > 04 Analytics > 06 Logistic Regression).
- Follow the steps outlines by the instructor.

Ex1: Use a logistic regression to predict Titanic survival.

Some hints

- Convert the numeric variable `Survived` to nominal level, in order to convince Knime to do classification.
- Drop `Name` in order to keep stuff simple.
- You might want to exclude missing values.
- For a Kaggle submission, only save the variables `PassengerID` and `Survived` (predicted); these two columns need be saved as CSV.

Solution in Knime

Get Knime Workflow file

Solution in R

Check out this post.

Tree model

GP1

- Let’s work on this example; find it here in your Knime Explorer: Examples > 04 Analytics > 01 Decision Tree.

Ex1

- Build a classification tree for the Titanic data set.
- Submit it to Kaggle, and check your score.

Solution via Knime

Check out this workflow.

Solution in R

Check out this post.

Boosted Trees

GP1

- Let’s work on this example; find it here in your Knime Explorer: Examples > 04 Analytics > 04_Classification and Predictive Modelling > 05 Gradient Boosted Trees.

Ex1

Use the following tuning parameters:

- max depth: 4
- Learning rate: 0.1

Solution via Knime

Check out this workflow.

Solution via R

Check out this post.

Random Forest 1, simple

GP1

- Let’s work on this example; find it here in your Knime Explorer: `Examples > 04_Analytics > 13_Meta_Learning > 02_Learning_a_Random_Forest`
- You may want tot check out the Knime course tapping into this topic.

Ex1

Use the following tuning parameters:

- 500 trees
- max depth: 10
- min. node size: 2
- Build the respective model for the Titanic data set; `Survived` is the outcome.
- Submit it to Kaggle, and check your score.

Feel free to check out e.g., this video on how to use Random Forests models for the Titanic Kaggle competition.

Solution via Knime

Check out this workflow.

Solution via R

Check out this post.

Random Forest 2 with parameter tuning

GP1

- Let's work on this example; find it here in your Knime Explorer: `Examples > 04_Analytics > 11_Optimization > 06_Parameter_Optimization_two_examples`
- Beware, execution of the second workflow may take some time, "computationally expensive", as they say.

Ex1

Optimize the following tuning parameters:

- Number of variables to consider per tree `columnAbsolutePerTree` (also known as `mtry`): `min=1, max=5`; step size is 1.

Check out this workflow.

Solution via Knime

Check out this workflow.

Random Forest 3, parameter tuning and cross validation

This time we'll add cross validation to the menu.

GP1

- Let's work on this example; find it here in your Knime Explorer: `Examples > 04_Analytics > 11_Optimization > 01_Cross_Validation_with_SVM`

Solution via Knime

Check out this workflow.

Solution via R

Check out this post.

BONUS: Boosting including tuning and crossvalidation via R

Check out this post.

Outro

That was ist, folks! I hope you enjoyed our journey. There's way more to explore; check out the resources for more.