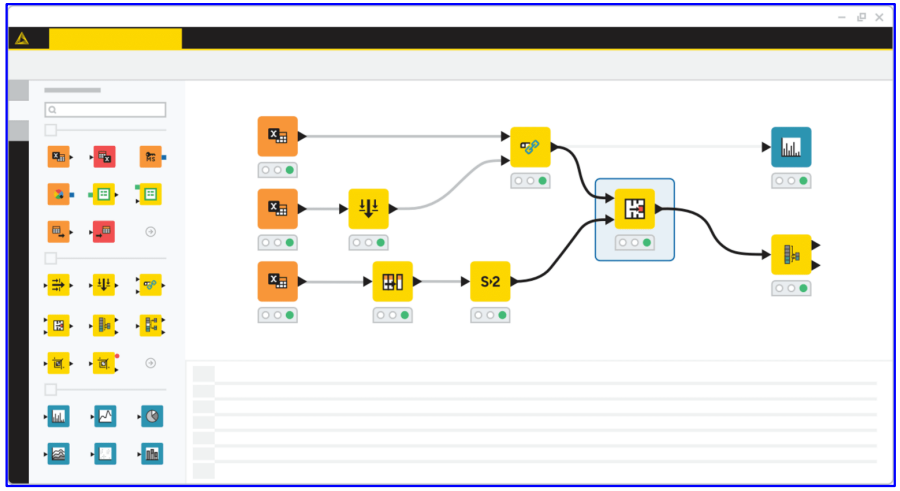
KNIME

Credit to TA.T

Outline

- What is KNIME?
- How to use KNIME?
- Download & install
- Create workflow
- KNIME Basics: Nodes, Colors, Ports, and Status
- LABs Overview
- LAB 1: Classification
- LAB 2: Regression



What is KNIME

KNIME (Konstanz Information Miner) is a visual, drag-and-drop platform for data analytics and machine learning that lets you build workflows without heavy coding.

- **Visual workflow tool:** connect “nodes” to create a data pipeline (like a flowchart).
- **End-to-end analytics:** import data → clean/transform → explore → train models → evaluate → export results.
- **Low-code / no-code friendly:** great for beginners, but can also integrate Python/R/SQL when needed.
- **Reproducible workflows:** your steps are saved as a workflow, so you can rerun and share the same process.
- **Extensible:** lots of built-in nodes + community extensions for ML, text mining, and more.

How to use KNIME?

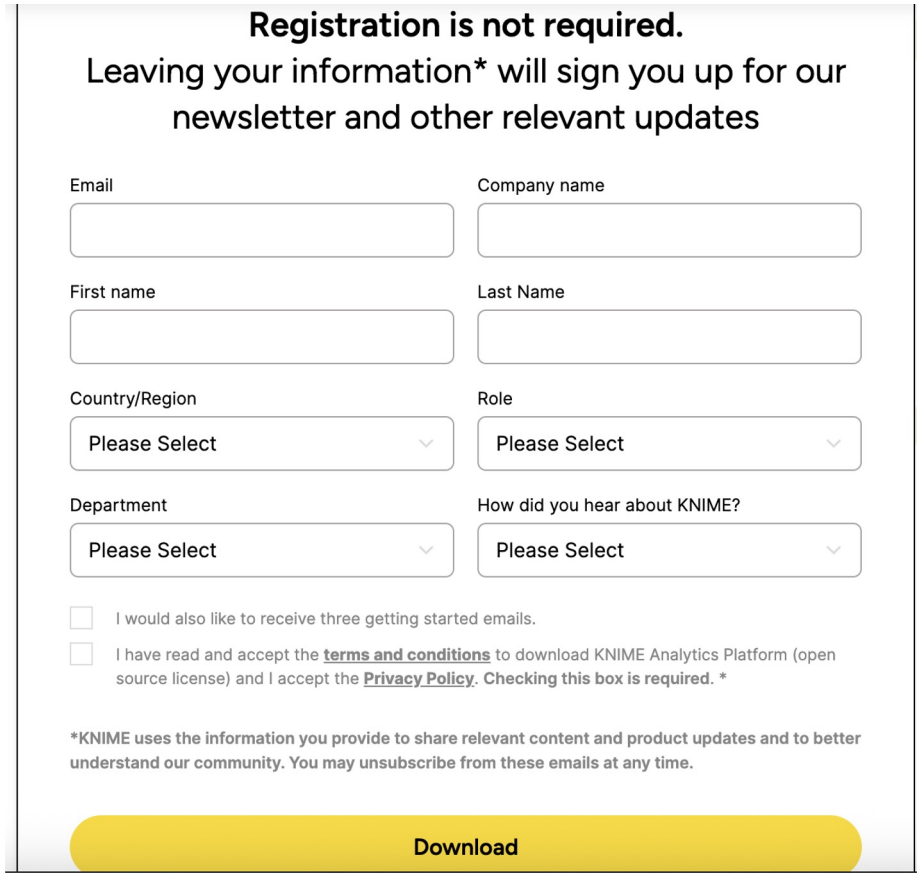
Local (Free)

- **KNIME Analytics Platform (Desktop)** = runs on your own computer
- Free and open-source (use it without paying).

Cloud / Collaboration (Paid)

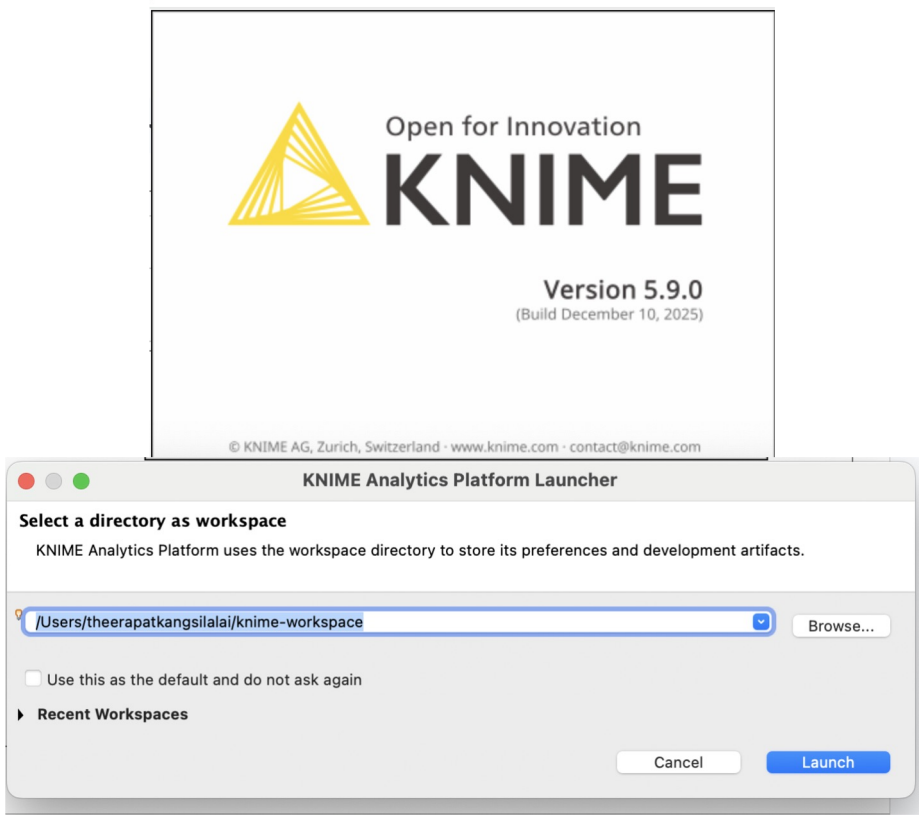
- KNIME Team plans add private collaboration and managed execution features.
- KNIME Business Hub: enterprise offering (SaaS or self-hosted); pricing is on request.

Download KNIME (LINK)



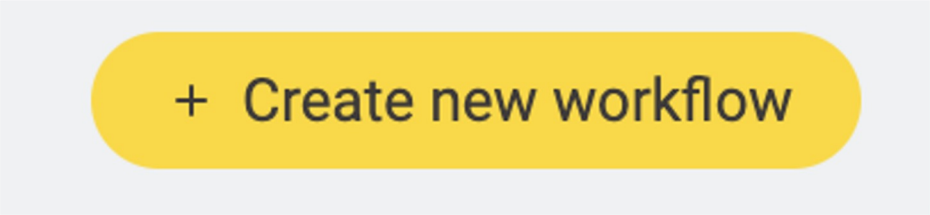
- Download KNIME Analytics Platform (Desktop) from the official KNIME website.
- **Install and launch the application.**
- Registration is not required to use KNIME Desktop for most labs (you can build and run workflows locally).
- In this course, **you will need to register/sign in later** to access a community extension node.

Open KNIME and select you folder



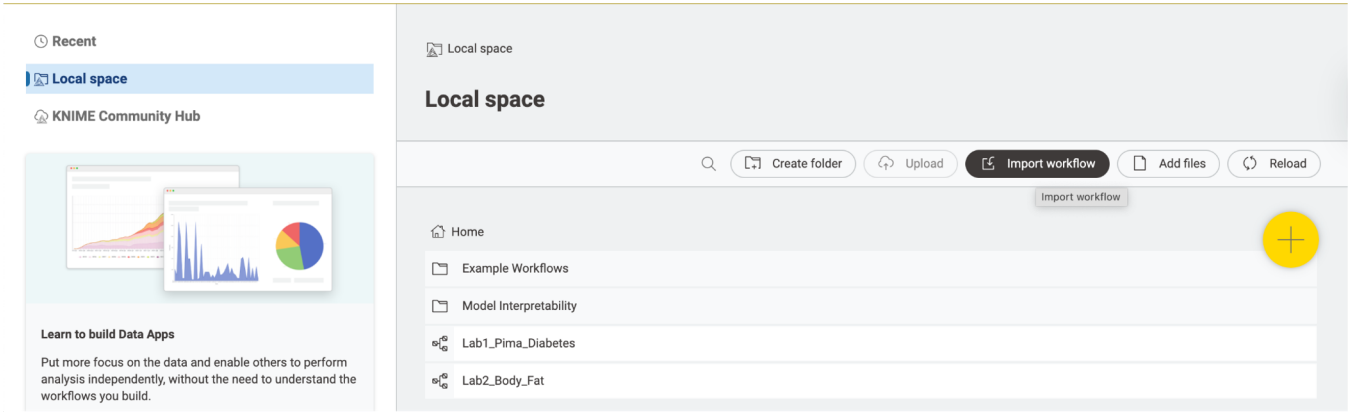
Then launch

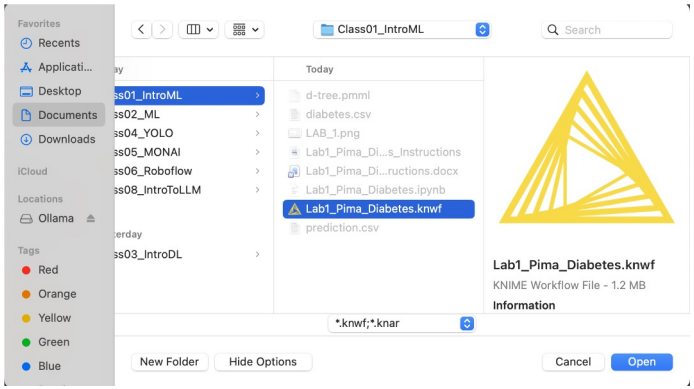
Create workflow



You can create your workflow **by clicking this button**.

Or import the existed workflow into your workspace in `**Local space**` tab.

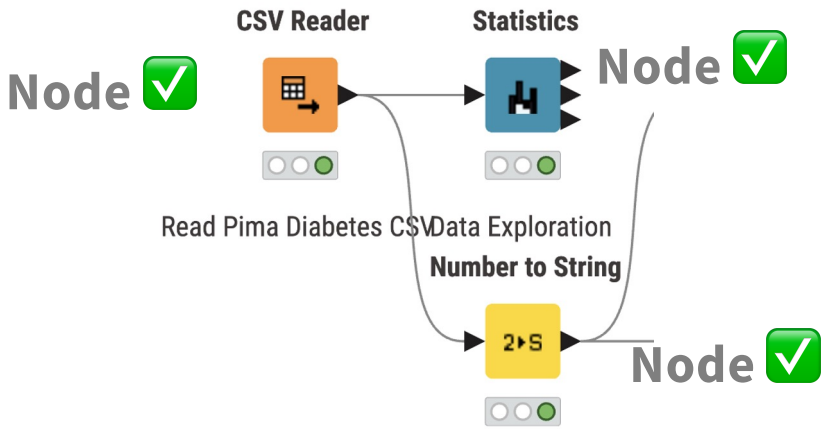




KNIME Basics: Nodes, Colors, Ports, and Status

What is a "Node"?

- A node is one step in your workflow (e.g., read data, clean data, train a model, evaluate results).
- Nodes are shown as colored boxes with input/output ports and a status indicator.



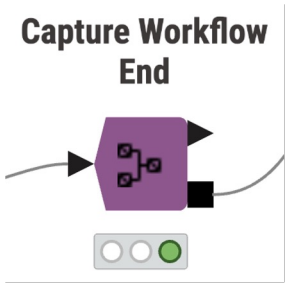
KNIME Basics: Nodes, Colors, Ports, and Status

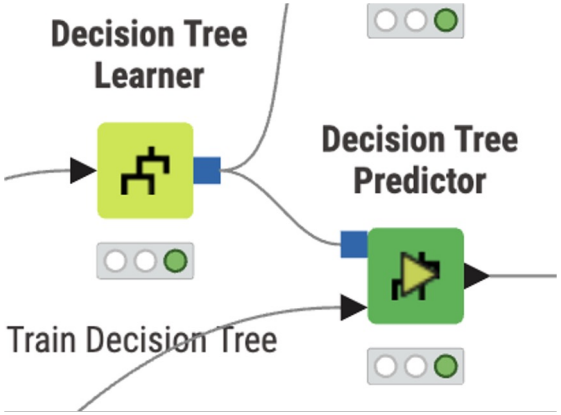
KNIME uses color as a quick “category hint” (useful for scanning a workflow; not a strict rule):

- **Orange:** Readers (import data)
- **Red:** Writers (export/save results)
- **Yellow:** Data manipulation / transformation
- **Light Green:** Model learners (train models)
- **Dark Green:** Predictors (apply models)
- **Blue:** Visualization
- **Light Blue:** Loop / flow control
- **Brown:** Utility
- **Gray:** Components (a packaged mini-workflow)

KNIME Basics: Nodes, Colors, Ports, and Status

- **Triangle ports (Data Table):** pass tabular data (rows/columns) between nodes.
- **Square ports (Model / Special objects):** pass objects like **models (PMML)** or other non-table objects.
- **Blue square** commonly indicates a **model/PMML** connection between learner → predictor.





KNIME Basics: Nodes, Colors, Ports, and Status

The “Three Circles” under each node (Node Status / Traffic Light)

- **Red** = not configured (new node / missing settings)
- **Yellow** = configured (settings done, not executed yet)
- **Green** = executed successfully

LABs Overview

Lab 1: Build your first end-to-end KNIME workflow (data → prep → model → evaluation) with **Classification** problem.

Lab 2: Extend the workflow with **feature selection** and **Regression** problem.

LAB 1: Classification

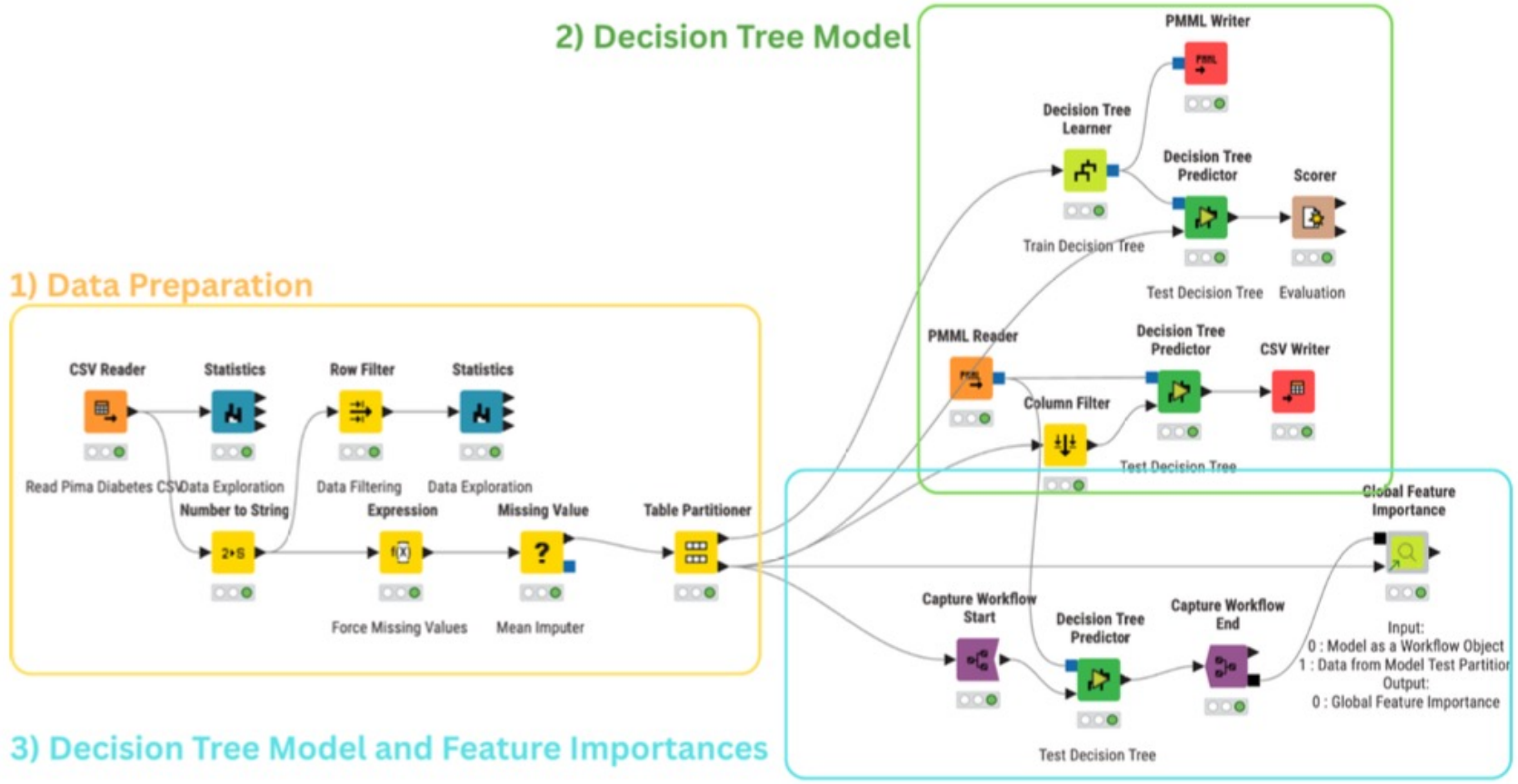
Data Description

- **Dataset:** Pima Indians Diabetes dataset
- **Records:** 768 patients (female, Pima Indian heritage, age ≥ 21)
- **Features (inputs):** Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DiabetesPedigreeFunction, Age
- **Target (label): Outcome** (binary)
    - 1 = diabetes, 0 = not diabetes

What We Will Do in KNIME

- **Load + inspect** the dataset (basic statistics, spot suspicious values like zeros)
- **Prepare the target** (convert Outcome to a nominal class column)
- **Handle "0" medical values** (demonstrate dropping rows vs converting to missing + imputing)
- **Split train/test** (e.g., 70/30, stratified, fixed seed for reproducibility)
- **Train a Decision Tree** classifier and **evaluate** with standard metrics (confusion matrix/ROC via Scorer)
- **Export and reuse the model** (PMML), generate **predictions.csv**, and explore **Global Feature Importance**

LAB 1: Classification



LAB 2: Regression

Data Description

- **Dataset:** Body Fat Prediction dataset (anthropometric measurements)
- **Records:** 252 adult male subjects (each row = one subject)
- **Target (to predict): BodyFat (%)**
- **Features (examples):** Age, Weight, Height, and multiple body circumferences (Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist)
- **Important note:** The dataset includes **Density**, but we **remove Density** because it is strongly related to BodyFat and acts like a "shortcut" feature (unrealistic in typical settings).

What We Will Do in KNIME

- **Load + inspect** the dataset (CSV Reader, Statistics)
- **Remove Density** using Column Filter (simulate a realistic measurement scenario)
- **Split data** into **Dev/Test = 80/20** with a fixed random seed (reproducibility)
- Build a **baseline Linear Regression** model and evaluate on Test using **Numeric Scorer** (report at least **RMSE**, optionally R^2)
- Run **wrapper feature selection** (train/validation split inside Dev), select the best feature subset by **minimizing validation RMSE**
- Apply **standardization (z-score)** and report the **final Test RMSE** using the selected features

LAB 2: Regression