Big Data Tools: KNIME, Spark, and Databricks

Lecture 6: Scalable Data Science with Free Tools (MSc Data Science)

Prof. Asc. Endri Raco, Ph.D. and AI Team

Department of Mathematical Engineering, Polytechnic University of Tirana

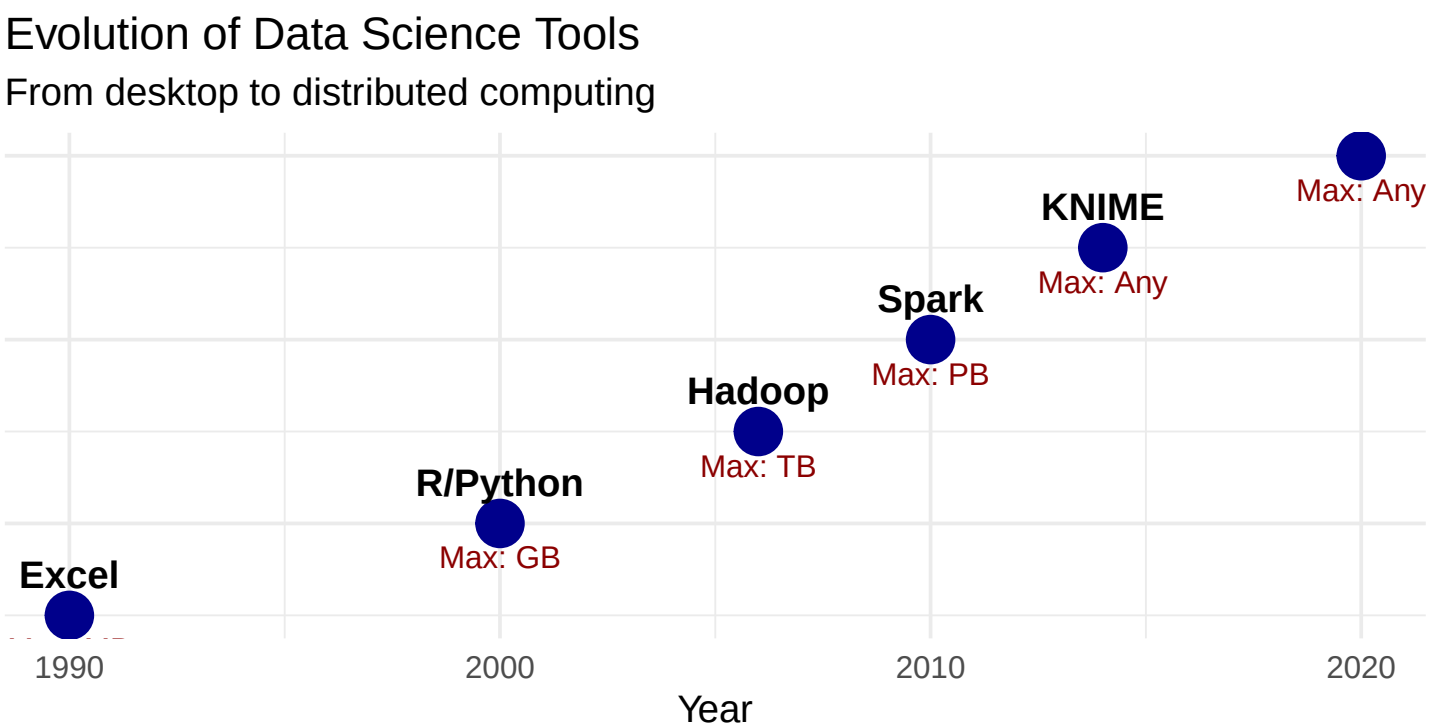
November 2025

Section 1

Part I: The Big Data Challenge and KNIME Workflows

Slide 1: The Big Data Revolution

Evolution of Data Science Tools:



Slide 2: Why Traditional Tools Break Down

The Scaling Problem:

Data Size	Tool	Processing Time	Memory
1 GB	R (laptop)	Minutes	8 GB RAM
10 GB	R (laptop)	Hours	32 GB RAM
100 GB	R (laptop)	**Crash!**	Not enough
100 GB	Spark (cluster)	Minutes	Distributed
1 TB	Spark (cluster)	Hours	Distributed

Key Insight: When data doesn't fit in memory, you need distributed computing

Slide 3: Lecture 6 Structure and Tools

Three Free Tools for Big Data:

1. **KNIME Analytics (Slides 1-40):**
   - Visual workflow designer (no-code)
   - Runs on laptop
   - Perfect for learning and prototyping
2. **Apache Spark + Databricks CE (Slides 41-100):**
   - Distributed computing framework
   - Free cloud tier (15 GB RAM)
   - Industry standard
3. **Integration (Slides 101-150):**
   - Combining KNIME, R, and Python
   - Deployment strategies

Slide 4: What is KNIME?

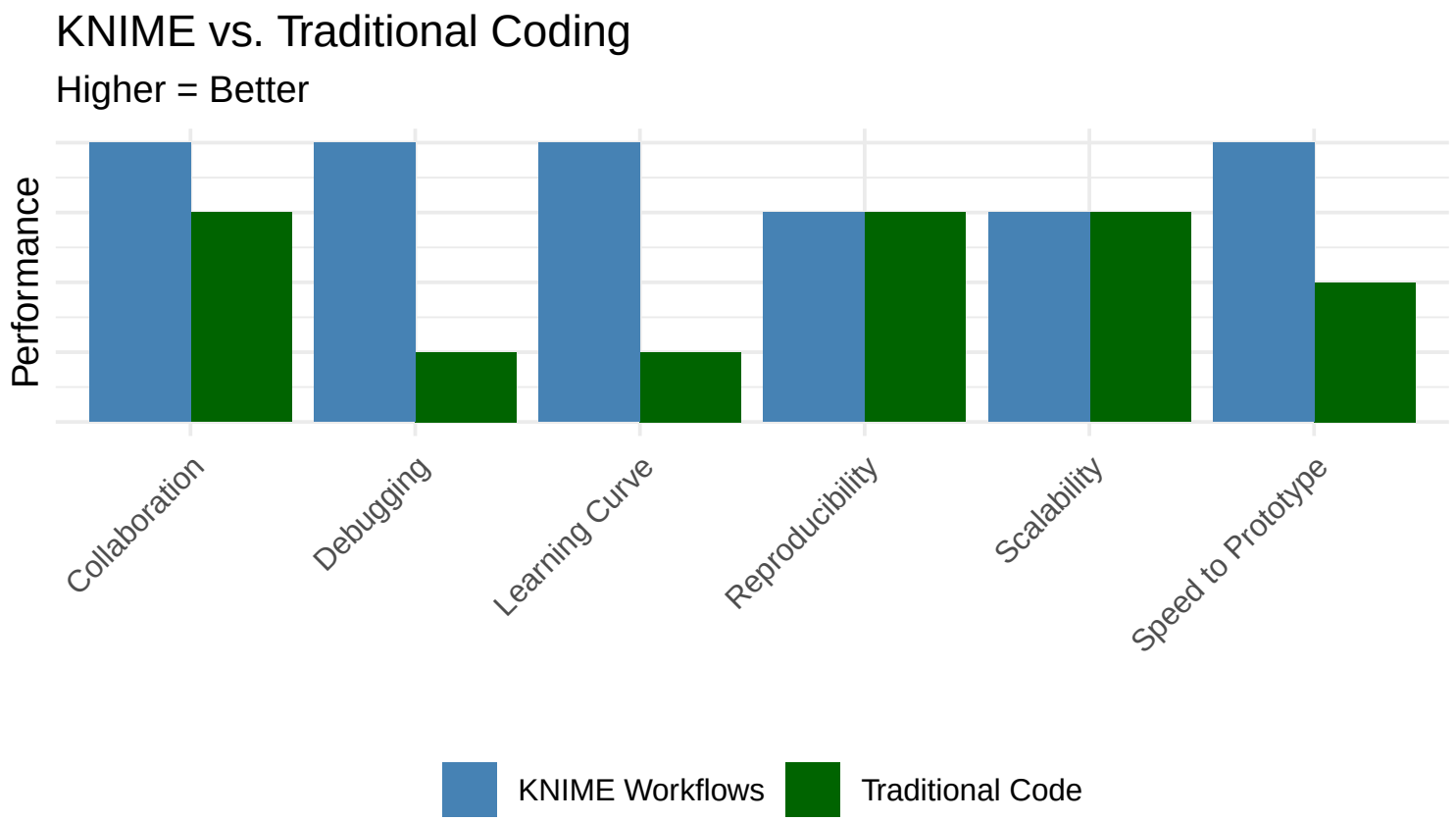
KNIME Analytics Platform:

- **K**onstanz **N**formation **M**iner
- Open-source, visual workflow tool
- Drag-and-drop interface (no coding required!)
- 2000+ pre-built nodes (operations)

Key Advantages:

- Free and open-source
- Runs on Windows, Mac, Linux
- Integrates R, Python, SQL, Spark
- Visual pipelines (easy to understand and share)
- Production-ready (can deploy workflows)

Slide 5: KNIME vs. Traditional Coding



Slide 6: Installing KNIME Analytics Platform

Installation Steps:

1. Visit: **https://www.knime.com/downloads**
2. Download KNIME Analytics Platform (free)
3. Install (no license needed)
4. Launch KNIME

System Requirements:

- Windows 10+, macOS 10.15+, or Linux
- 4 GB RAM minimum (8 GB recommended)
- 2 GB disk space
- Java 11+ (included in installer)

First Launch: Creates workspace folder for your workflows

Slide 7: KNIME Interface Overview

Main Components:

1. **Node Repository (Left):** Library of 2000+ operations
2. **Workflow Canvas (Center):** Drag nodes here
3. **Workflow Coach (Right):** Suggests next steps
4. **Description (Bottom):** Node documentation
5. **Console (Bottom):** Execution messages

Key Terms:

- **Node:** Single operation (read file, filter, model)
- **Workflow:** Connected nodes (complete pipeline)
- **Port:** Connection point (triangle = data table)
- **Configure:** Set node parameters (double-click)
- **Execute:** Run node (right-click $\rightarrow$ Execute)

Slide 8: Your First KNIME Workflow - Hello Data

Simple 3-Node Workflow:

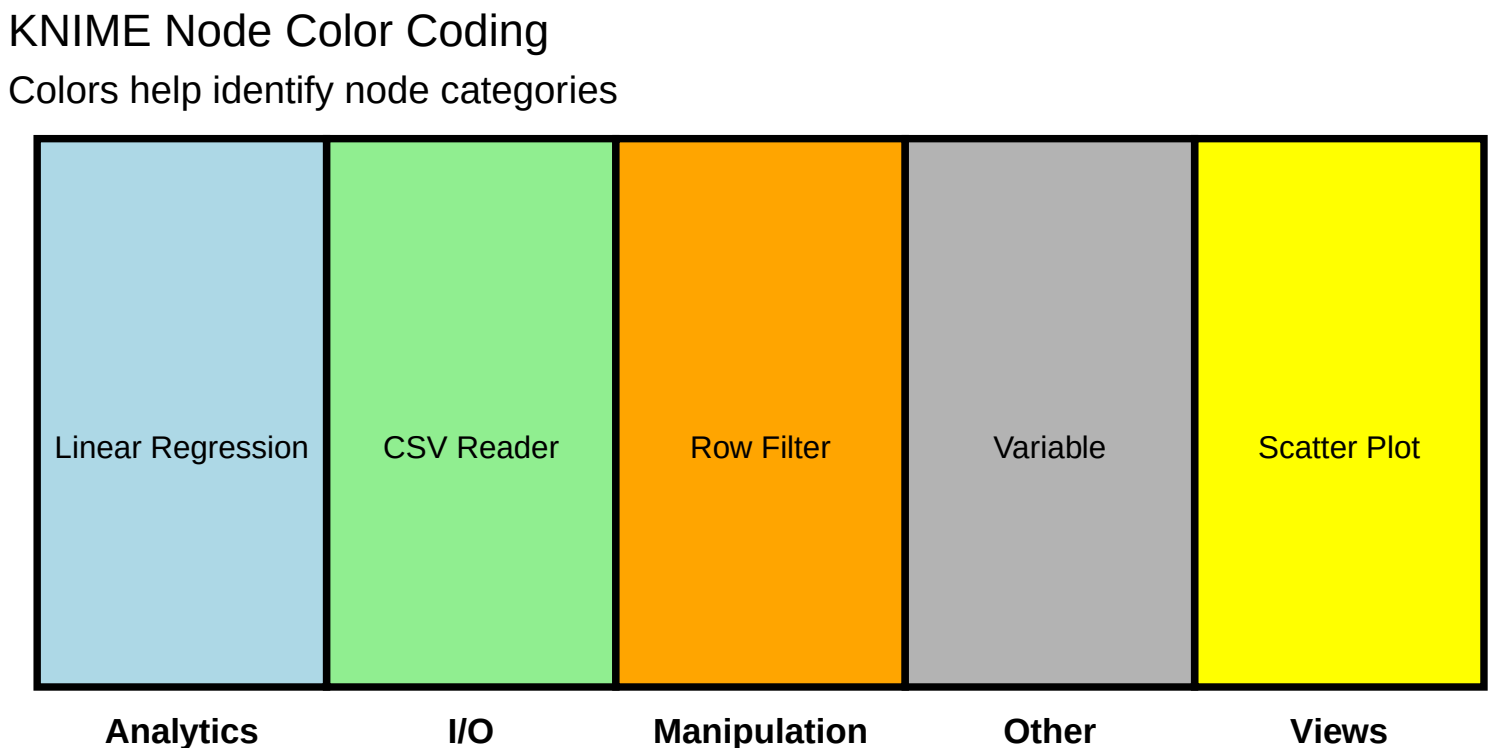
```
[Data Generator] → [Row Filter] → [Table View]
```

Steps:

1. **Node Repository → Manipulation → Row → Data Generator**
   - Drag to canvas
   - Configure: 100 rows
   - Execute (green light = success)
2. **Manipulation → Row → Row Filter**
   - Drag to canvas, connect to Data Generator
   - Configure: Keep rows where Column0 $>$ 50
   - Execute
3. **Views → Table View**
   - Connect and execute
   - View results

Slide 9: KNIME Node Types - Color Coding

Node Colors Indicate Function:



Port Shapes: Triangle = data table, Square = model, Circle = other

Slide 10: Reading Data in KNIME

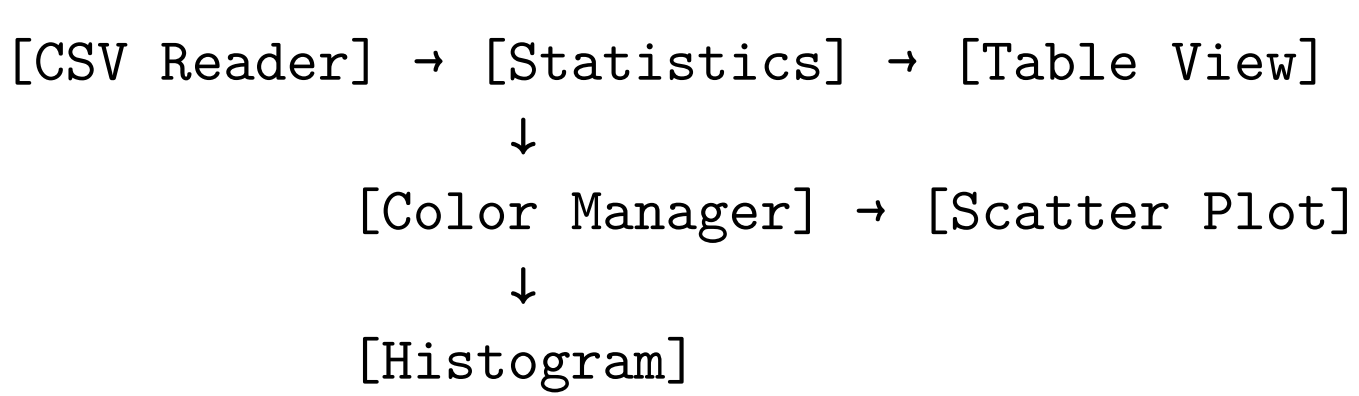
Common Data Sources:

1. **CSV Reader** (most common)
   - File → CSV Reader
   - Browse to file
   - Auto-detects delimiter, headers
2. **Excel Reader**
   - Reads .xlsx files
   - Select specific sheets
3. **Database Connector**
   - MySQL, PostgreSQL, SQLite
   - Execute SQL queries
4. **File Reader** (generic)
   - Auto-detects format

Demo: Read iris.csv dataset

Slide 11: Data Exploration Nodes

Essential Exploration Nodes:



Key Nodes:

- **Statistics:** Mean, median, min, max per column
- **Table View:** Browse data (like View() in R)
- **Histogram:** Distribution of numeric columns
- **Scatter Plot:** Relationships between variables
- **Box Plot:** Outlier detection
- **Missing Value:** Check for NAs

Slide 12: Data Cleaning Workflow

Common Cleaning Operations:

```
[CSV Reader]
    ↓
[Column Filter] (remove unwanted columns)
    ↓
[Row Filter] (remove outliers/bad data)
    ↓
[Missing Value] (handle NAs)
    ↓
[String Manipulation] (clean text)
    ↓
[Normalizer] (scale numeric features)
    ↓
[Column Rename] (standardize names)
```

Best Practice: Chain nodes to create reproducible cleaning pipeline

Slide 13: Feature Engineering in KNIME

Creating New Features:

1. **Math Formula Node:**
   - Create derived columns
   - Example: `$price$ / $area$ = price_per_sqm`
2. **String Manipulation:**
   - Extract substrings
   - Convert case
   - Replace patterns
3. **Rule Engine:**
   - If-then-else logic
   - Example: `$age$ > 65 => "Senior"`
4. **Java Snippet:**
   - Custom code for complex operations

Slide 14: Partitioning Data - Train/Test Split

Node: Partitioning

```
[CSV Reader]
    ↓
[Partitioning] (80% train / 20% test)
    ↓ (two outputs)
[Training Data]    [Test Data]
```

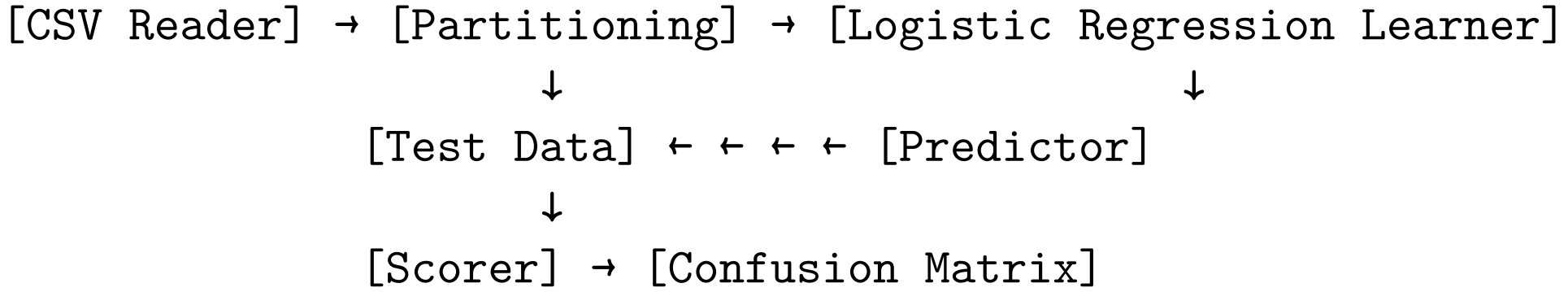
Configuration:

- **Relative:** 80% / 20%
- **Absolute:** First 1000 rows / rest
- **Stratified:** Preserve class distribution
- **Random Sampling:** With seed for reproducibility

Important: Set random seed for reproducible splits!

Slide 15: End-to-End Classification Workflow - Overview

Complete Iris Classification Pipeline:



Workflow Steps:

1. Load iris data
2. Split 80/20
3. Train logistic regression on training set
4. Predict on test set
5. Evaluate with confusion matrix

Slide 16: Step 1 - Load Iris Data

Node: CSV Reader

Configuration:

- Browse to `iris.csv`
- Check "Has Column Headers"
- Check "Has Row IDs"
- Click OK

Execute and View:

- Right-click → Execute
- Right-click → Output → Data Table
- Verify: 150 rows, 5 columns
- Species column should be categorical

If Species is not categorical: Add **String to Number** node

Slide 17: Step 2 - Partition Data

Node: Partitioning

Configuration:

1. Drag **Manipulation → Row → Partitioning**
2. Connect to CSV Reader
3. Configure:
   - Relative: 80% top / 20% bottom
   - Stratified sampling: Check
   - Column: Species
   - Random seed: 123

Why Stratified? Preserves class distribution in both sets

Execute: Node should show two output ports (top = train, bottom = test)

Slide 18: Step 3 - Train Logistic Regression

Node: Logistic Regression Learner

Path: Analytics → Mining → Logistic Regression Learner

Configuration:

1. Connect top port of Partitioning node
2. Double-click to configure:
   - Target column: Species
   - Feature columns: Select all numeric columns
   - Solver: Default (IRLS)

Execute:

- Green light = model trained successfully
- Model object stored in output port (square shape)

Slide 19: Step 4 - Make Predictions

Node: Predictor

Configuration:

1. Drag **Analytics → Mining → Predictor**
2. Connect TWO inputs:
   - Model from Logistic Regression Learner (square port)
   - Test data from Partitioning (bottom triangle port)
3. Configure:
   - Append columns with suffix: _predicted
   - Include probabilities: Check

Execute: Adds prediction columns to test data

Slide 20: Step 5 - Evaluate Model

Nodes: Scorer + Confusion Matrix

```
[Predictor] → [Scorer] → [Confusion Matrix]
```

Scorer Node:

- Analytics → Mining → Scorer
- Connect to Predictor output
- Configure:
  - First column: Species (actual)
  - Second column: Species_predicted
- Execute

Confusion Matrix:

- Views → Confusion Matrix
- Connect to Scorer
- Execute and view results

Slide 21: Understanding KNIME Output

Scorer Output Statistics:

- **Accuracy:** Overall correctness
- **Precision:** True positives / (TP + FP)
- **Recall:** True positives / (TP + FN)
- **F1-Score:** Harmonic mean of precision and recall

Confusion Matrix:

	Predicted Setosa	Predicted Versicolor	Predicted Virg
Actual Setosa	TP	FP	FP
Actual Versicolor	FN	TP	FP
Actual Virginica	FN	FN	TP

Slide 22: Saving Your Workflow

Save Workflow:

1. File → Save As
2. Choose location
3. Name: `Iris_Classification`
4. Creates `.knwf` file

Export Workflow:

1. File → Export KNIME Workflow
2. Creates portable `.knwf` archive
3. Can share with others

Best Practice:

- Save frequently
- Use version control (Git-friendly)
- Add annotations (Edit → Workflow Annotations)

Slide 23: Adding Documentation to Workflows

Making Workflows Understandable:

1. **Node Descriptions:**
   - Right-click node → Edit Node Description
   - Explain what this node does
2. **Workflow Annotations:**
   - Right-click canvas → Workflow Annotation
   - Add text boxes explaining sections
3. **Meta Nodes:**
   - Group related nodes
   - Right-click nodes → Create Meta Node
   - Collapse complex sections

Good Practice: Document as you build, not after!

Slide 24: KNIME Components - Reusable Modules

What are Components?

- Reusable sub-workflows
- Encapsulate complex logic
- Share across projects

Create Component:

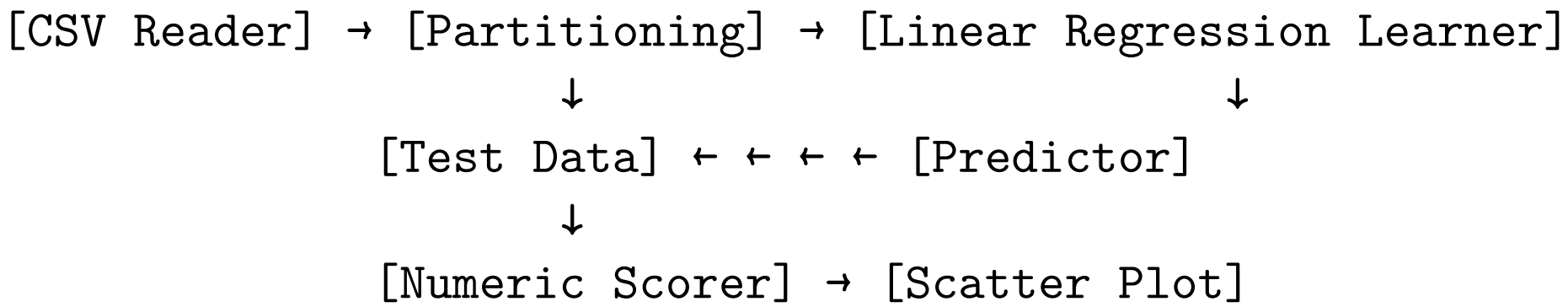
1. Select multiple nodes
2. Right-click $\rightarrow$ Create Component
3. Name it (e.g., "Data Cleaning")
4. Configure inputs/outputs

Use Cases:

- Standard data preprocessing
- Custom visualization
- Feature engineering pipelines

Slide 25: Regression Workflow in KNIME

Predicting House Prices:



Key Differences from Classification:

- **Learner:** Linear Regression (not Logistic)
- **Scorer:** Numeric Scorer (not Classification Scorer)
- **Metrics:** RMSE, MAE, R^2 (not accuracy)

Slide 26: Regression Evaluation Nodes

Numeric Scorer Output:

```
[Predictor] → [Numeric Scorer]
                      ↓
              Statistics Output:
              - Mean Absolute Error (MAE)
              - Root Mean Squared Error (RMSE)
              - R² (coefficient of determination)
              - Mean Signed Difference
```

Visualization:

- **Scatter Plot:** Actual vs. Predicted
  - Perfect predictions = diagonal line
  - Points above = over-predictions
  - Points below = under-predictions

Slide 27: Advanced Nodes - Cross-Validation

X-Partitioner + X-Aggregator:

```
[Data] → [X-Partitioner (k=5)]
              ↓ (loop)
         [Learner] → [Predictor] → [Scorer]
              ↓ (collect results)
         [X-Aggregator]
              ↓
         [Mean Accuracy across folds]
```

Configuration:

- X-Partitioner: Set number of folds (k=5 or k=10)
- Stratified: Preserve class distribution
- X-Aggregator: Calculates average performance

Slide 28: Parameter Optimization in KNIME

Parameter Optimization Loop:

```
[Data] → [Parameter Optimization Loop Start]
              ↓
         [Learner with varying parameters]
              ↓
         [Cross-validation]
              ↓
         [Parameter Optimization Loop End]
              ↓
         [Best parameters selected]
```

Use Case: Find optimal hyperparameters (similar to grid search in R)

Example: Optimize regularization in Logistic Regression

Slide 29: Exporting Models from KNIME

Deployment Options:

1. **PMML (Predictive Model Markup Language):**
   - Model → PMML Writer
   - Export model in standard format
   - Import into R, Python, Java
2. **Python Node:**
   - Convert KNIME model to Python
   - Deploy in Python environments
3. **REST Service:**
   - KNIME Server (commercial)
   - Expose workflow as API
4. **Batch Scoring:**
   - Save workflow
   - Run headless: `knime -consoleLog -nosplash -application org.knime.product.KNIME_BATCH_APPLICATION`

Slide 30: KNIME Hub - Share and Discover

KNIME Hub: hub.knime.com

Features:

- **Public Workflows:** Download examples
- **Components:** Reusable building blocks
- **Extensions:** Additional node packages
- **Community:** Ask questions, share solutions

Popular Extensions:

- **Deep Learning:** TensorFlow, Keras integration
- **Text Processing:** NLP nodes
- **Big Data:** Spark, Hadoop connectors
- **Time Series:** Specialized forecasting nodes

Next Lecture Section: Introduction to Spark and Databricks (Slides 31-60)