

Progetto di architetture dati: Inspector Gadget behaviours

Tiberio Falsiroli(874971), Andrea Premate (829777)

Presentation Outline

- **Introduction**
- **Dataset and workflow structure**
- **Main Process Workflow**
- **Inspector Gadget Monitor**

— — —

Introduction

Inspector Gadget

KNIME

Obbiettivi

Introduction

Inspector Gadget

- Framework che semplifica la stratificazione delle capacità di monitoraggio e debug su un Dataflow Engine esistente.
- Fornisce astrazioni per osservare i dati nel loro percorso attraverso il flusso, etichettare parti di questi dati, visualizzare tag, scambiare messaggi tra coppie di punti di osservazione e/o con un nodo coordinatore centrale.

Introduction



- Tool che permette in maniera grafica ed intuitiva di aggiungere e togliere nodi per il monitoring, la modifica, la rimozione l'unione e l'analisi di un insieme di dati.
- Nel progetto sono stati utilizzati sia nodi predefiniti in KNIME per azioni specifiche, che create funzionalità da zero tramite dei nodi Java Snippet, in cui è possibile inserire del codice totalmente customizzabile.

Introduction

— — —

Obbiettivi

Implementare una concretizzazione del framework Inspector Gadget applicandola ad un workflow esistente (creato da noi), allo scopo di:

- identificare dati/operazioni che causano crash
- assicurare consistenza dei dati
- garantire l'healthness del workflow

Dataset and workflow structure

Dataset and workflow structure

Dataset

- Statlog (Shuttle) Dataset
- Dataset utilizzato per estrarre regole utili a determinare il tipo di atterraggio che un veicolo spaziale deve eseguire.
- 9 attributi numerici (es. tempo), la colonna target ha 7 possibili classi
- Dataset contaminato da alcune entries sporche.
- Il dataset viene diviso in training e test set per task di machine learning.

Data Set Characteristics:	Multivariate	Number of Instances:	58000	Area:	Physical
Attribute Characteristics:	Integer	Number of Attributes:	9	Date Donated	N/A
Associated Tasks:	Classification	Missing Values?	N/A	Number of Web Hits:	162502

Dataset and workflow structure

Workflow structure

Divisione in tre macro aree

- Main process workflow: contiene i nodi che implementano le operazioni principali che si vogliono svolgere in un processo. Nel nostro caso operazioni di scrittura/lettura del database e task di machine learning.
- Processing dei controlli: contiene i nodi che implementano la logica dei controlli. Layer alimentato dal layer sottostante(main process) per la realizzazione del layer sovrastante(inspector gadget monitor).
- Inspector Gadget Monitor: contiene i nodi che mostrano i risultati delle analisi fatte. Leggendo gli output si acquisiscono informazioni utili sulla qualità dei dati.

INSPECTOR GADGET MONITOR

Row Filter
Controllo sul numero di righe
(table-level integrity alerts)

Row Filter
Mostro righe etichettate
(data samples)

Scorer (deprecated)
score results
(trial runs)

Row Filter
dati con classi non riconoscibili
(row-level integrity alerts)

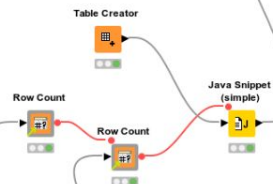
(Data summaries)

Statistics
Statistiche sui dati

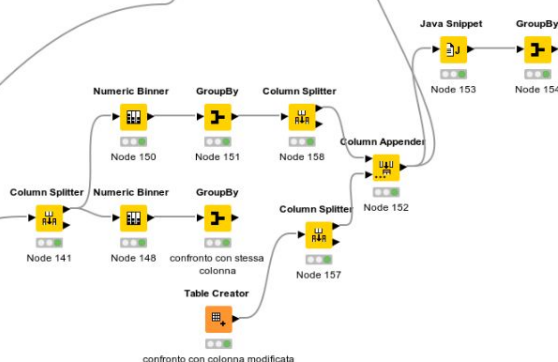
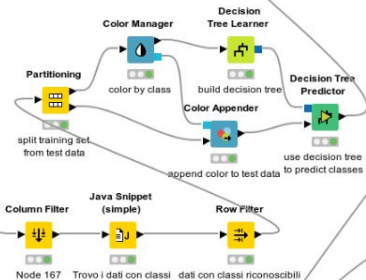
Kolmogorov-Smirnov
Test
Kolmogorov-Smirnov Test

Java Snippet
Hellinger distance
(used to quantify the similarity between
two probability distributions)

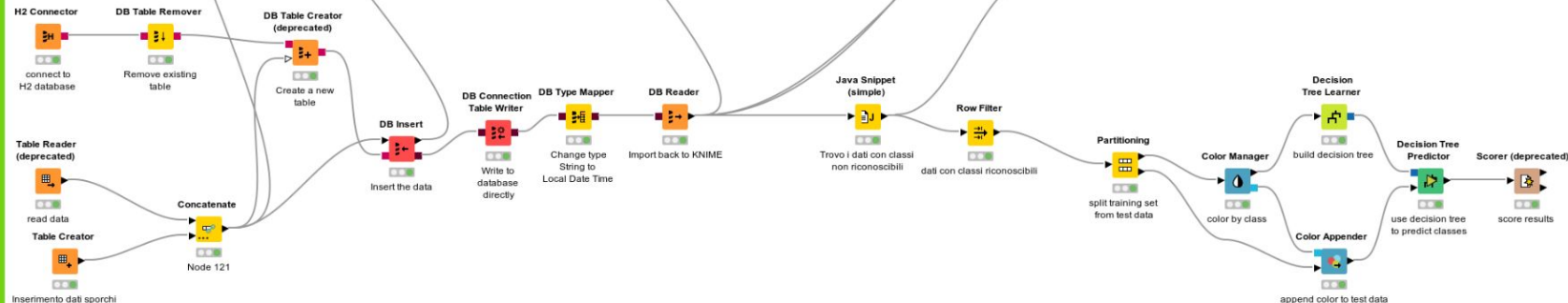
PROCESSING DEI CONTROLLI



Java Snippet (simple)
Etichetta a campione delle entry



MAIN WORKFLOW PROCESS



Main Process Workflow

DB Operations

Machine Learning Task

Main Process Workflow

— — — Intro

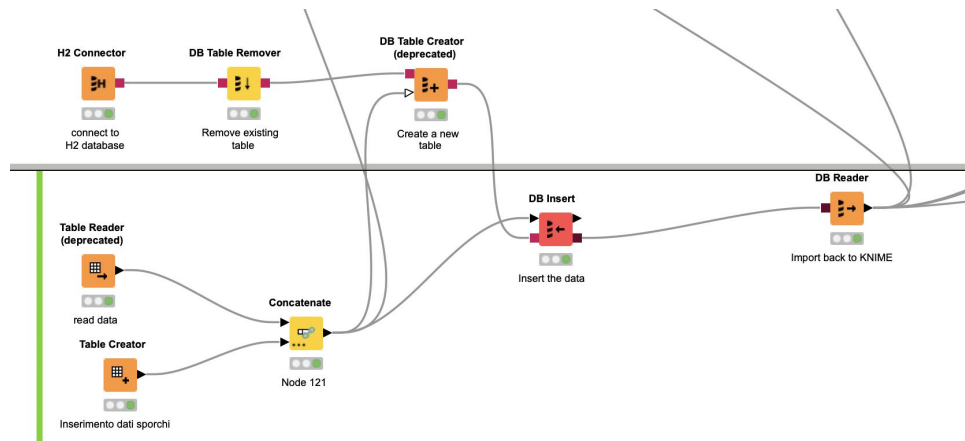
Racchiude al suo interno tutti i nodi necessari per svolgere il task principale, come la connessione al database, la creazione di tabelle e l'esecuzione di task di machine learning.

Main Process Workflow

DB Operations

In questa parte di workflow

- Viene instaurata la connessione col database
- Viene creata la tabella con i dati “sporchi” e inserita nel DB
- La tabella viene resa disponibile a tutti i nodi che ne hanno bisogno all’interno delle altre parti del workflow

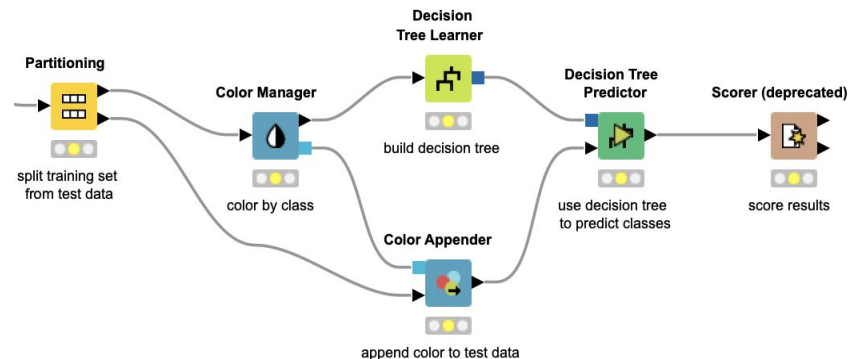


Main Process Workflow

Machine Learning Task

In questa parte di workflow è racchiuso il nostro task di machine learning

- Viene diviso il dataset in testset e trainset
- Viene costruito un decision tree che effettua una previsione sul test set
- Vengono mostrati i risultati e la bontà del nostro decision tree (confusion matrix, accuracy, recall...)



Inspector Gadget Monitor

Table-level
Integrity
Alerts

Data
samples

Trial Runs

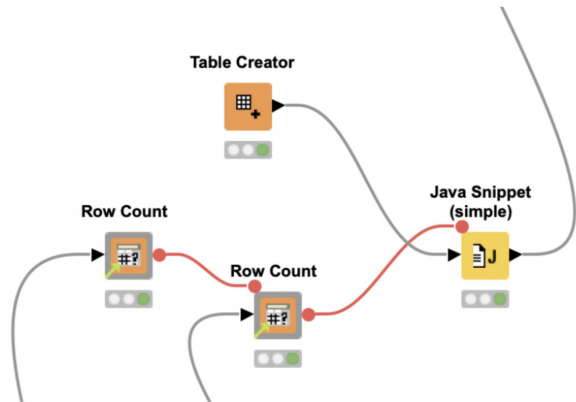
Row-level
Integrity
Alerts

Data
Summaries

Inspector Gadget Monitor

Table-level Integrity Alerts

- Lancia un alert quando un dataset presente in passaggi intermedi dell'elaborazione viola un certo vincolo.
- Dopo aver ricevuto in input il conteggio delle righe in due punti del workflow differenti, mette questi valori a confronto.
- La soluzione è facilmente modificabile e scalabile



Inspector Gadget Monitor

Row-level Integrity Alerts

- Lancia un alert ogni volta che un record viola un certo vincolo, ad esempio il campo X non deve essere nullo
- Impostando a priori in un Java Snippet le classi con cui il workflow si aspetta di lavorare, siamo in grado di individuare entry anomale e in seguito di isolarle e/o mostrarle nel monitor Inspector Gadget.
- A destra possiamo vedere il codice contenuto nel nostro java snippet

Method Body

```
Double bin;
if ($Col9$.equals("class0") || $Col9$.equals("class1") ||
$Col9$.equals("class2") || $Col9$.equals("class3") ||
$Col9$.equals("class4") || $Col9$.equals("class5") ||
$Col9$.equals("class6")) {
    bin=1.0;
}else{
    bin = 2.0;
}

return bin;
```

Inspector Gadget Monitor

Data samples

- Mostrare un sottoinsieme dei dati del workflow come sanity check per trovare dati sistematicamente errati, come ad esempio una colonna con solo valori nulli.
- Nel nostro workflow questo controllo `e reso possibile da un java snippet che racchiude un metodo (visionabile nella figura a destra)

Method Body

```
Double bin=0.0;
boolean val = new Random().nextInt(50)==0;

if(val){
    bin=1.0;
}

return bin;
```

Inspector Gadget Monitor

— — —

Trial Runs

- Eseguire il workflow su un piccolo sottoinsieme dei dati di input come veloce sanity check per vedere se avviene un crash o se si ottengono risultati coerenti con le aspettative.
- Utilizza un sample proveniente dalla parte di monitoring “Data samples”. Su tale campione viene quindi eseguito il task di classificazione come nel processo principale. Si ottiene un modello più semplice.

Trial Runs



Inspector Gadget Monitor

Data summaries

- Calcolare statistiche su dati (es. istogramma) su una run del workflow e confrontare con run precedenti.
- Trovare cambiamenti in distribuzioni dei dati per rilevare possibili errori di processing.

Statistics

Kolmogorov-Smirnov Test

Hellinger Distance

Inspector Gadget Monitor

Data summaries

Row Filter



assi non riconoscibili
I integrity alerts)

(Data summaries)

Statistics



Statistiche sui dati

Kolmogorov-Smirnov
Test

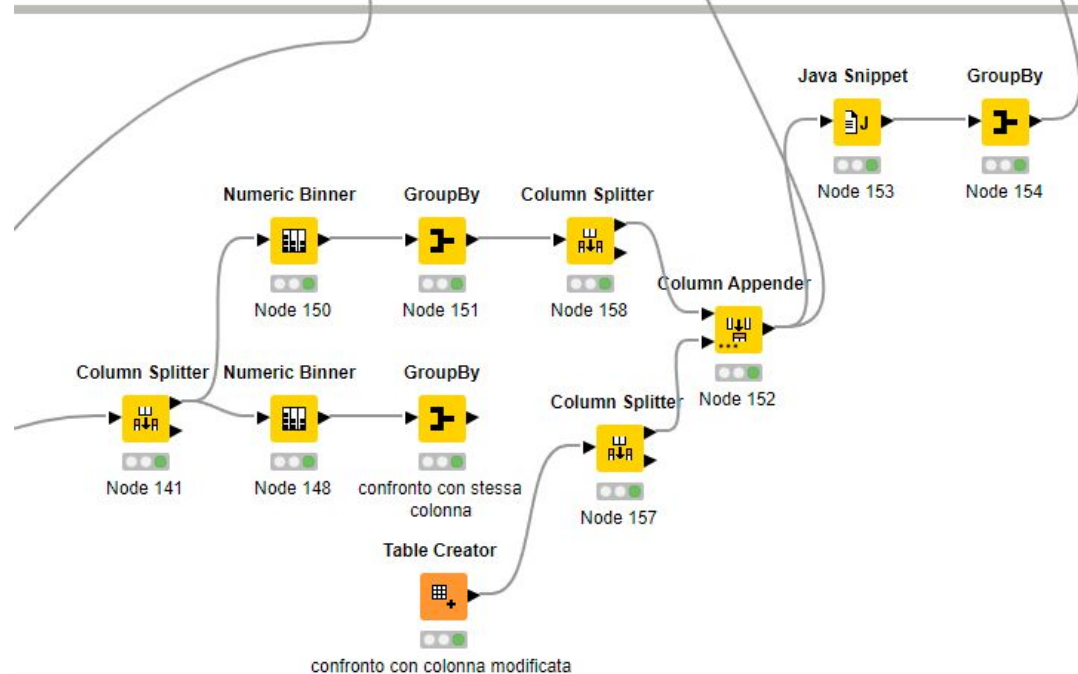


Kolmogorov-Smirnov Test

Java Snippet




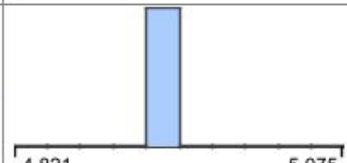
Hellinger distance
(used to quantify the similarity between
two probability distributions)



Inspector Gadget Monitor

Data summaries: Statistics

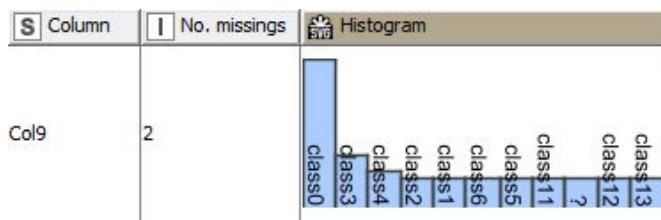
- Dopo lettura dei dati dal DB vengono calcolate statistiche relative ai dati stessi.
- Statistiche relative ad attributi numerici:

S	Column	D	Min	D	Max	D	Mean	D	Std....	D	Variance	D	Skewness	D	Kurtosis	D	Overall sum	I	No. miss...	I	I	I	I	I	I	Histogram
	Col0		27		126		48.239		12.237		149.753		2.181		6.51		2,798,173		0		0	0	0	?	...	
	Col1		-4,821		5,075		-0.019		77.953		6,076.722		6.439		2,647.637		-1,128		0		0	0	0	?	...	

Inspector Gadget Monitor

Data summaries: Statistics

- Statistiche relative ad attributi nominali



S Col9	I Count (Col9)	D Relative Frequency (Col9)
class0	45586	0.786
class3	8903	0.153
class4	3267	0.056
class2	171	0.003
class1	50	0.001
class6	13	0
class5	10	0
class11	3	0
?	2	0
class12	1	0
class13	1	0

Inspector Gadget Monitor

Data summaries: Kolmogorov-Smirnov Test

- Test statistico per verificare che due campioni di dati provengono dalla stessa distribuzione originaria.
- Viene calcolato il p-value che assieme al livello di significatività alfa (regolabile dall'utente) determina se l'ipotesi nulla H_0 (i due sample di dati provengono dalla stessa distribuzione) debba essere rifiutata o no.
- p-value: la probabilità, per una ipotesi supposta vera (ipotesi nulla), di ottenere risultati ugualmente o meno compatibili, di quelli osservati durante il test, con la suddetta ipotesi.

Inspector Gadget Monitor

— — —

Data summaries: Kolmogorov-Smirnov Test

Nel workflow vengono mostrati due esempi di funzionamento:

- Colonna del dataset confrontata con se stessa: i due sample provengono dalla stessa distribuzione di dati per qualsiasi alpha richiesto.
- Colonna del dataset confrontata con versione modificata di se stessa: tanto più cambia la versione modificata, tanto più si ottiene un p-value minore, quindi è più facile rigettare l'ipotesi nulla.

In un sistema reale si confronterebbero dataset, provenienti da misurazioni e processi analoghi, relativi ad esempio a diversi istanti di tempo.

Inspector Gadget Monitor

Data summaries: Hellinger Distance

- Utilizzata per quantificare la similarità tra due distribuzioni di probabilità.
- Date due distribuzioni di probabilità discrete $P = (p_1, \dots, p_k)$ e $Q = (q_1, \dots, q_k)$ è definita come:

$$H(P, Q) = \frac{1}{\sqrt{2}} \sqrt{\sum_{i=1}^k (\sqrt{p_i} - \sqrt{q_i})^2}$$

Inspector Gadget Monitor

— — — Data summaries: Hellinger Distance

- Implementata con Java Snippet.
- Stessi esempi di funzionamento di Kolmogorov-Smirnov Test:
 - Colonna confrontata con se stessa -> distanza 0
 - Colonna confrontata con v. modificata -> maggiori modifiche, maggiore distanza
- Discorso analogo su dataset simili relativi ad esempio a istanti di tempo diversi.

Inspector Gadget Monitor

Data summaries: Hellinger Distance

- Esempio di applicazione:
 1. Dataset corretto viene preso come standard di riferimento.
 2. Dataset simili analizzati sequenzialmente sono associabili ad una successione di valori di distanze.
 3. Alert se si ottiene la distanza max rispetto all'intero storico, o con tolleranza.
- **Variazione:** utilizzare più distribuzioni (differenti perché con semantica differente) di riferimento allo scopo di riconoscere la semantica dell'intera colonna.
- **Esempio:** due dataset in cui il tempo viene misurato in uno con settimane e nell'altro con giorni. Avendo due distribuzioni di riferimento (una per casistica) si può automaticamente dire se 5, specifico campo di un record, si riferisce a giorni o settimane (analizzando l'intera colonna).

Thank you for your attention