Politecnico di Torino

Pre-Lab 6

Business Intelligence per Big Data

Random Forest | **Ottimizzare parametri** | **SVM**

Reti Neurali | **Overfitting ed Underfitting**

Eleonora Poeta, Adriano De Cesare

Obiettivo 1

Random Forest

Tecnica di classificazione

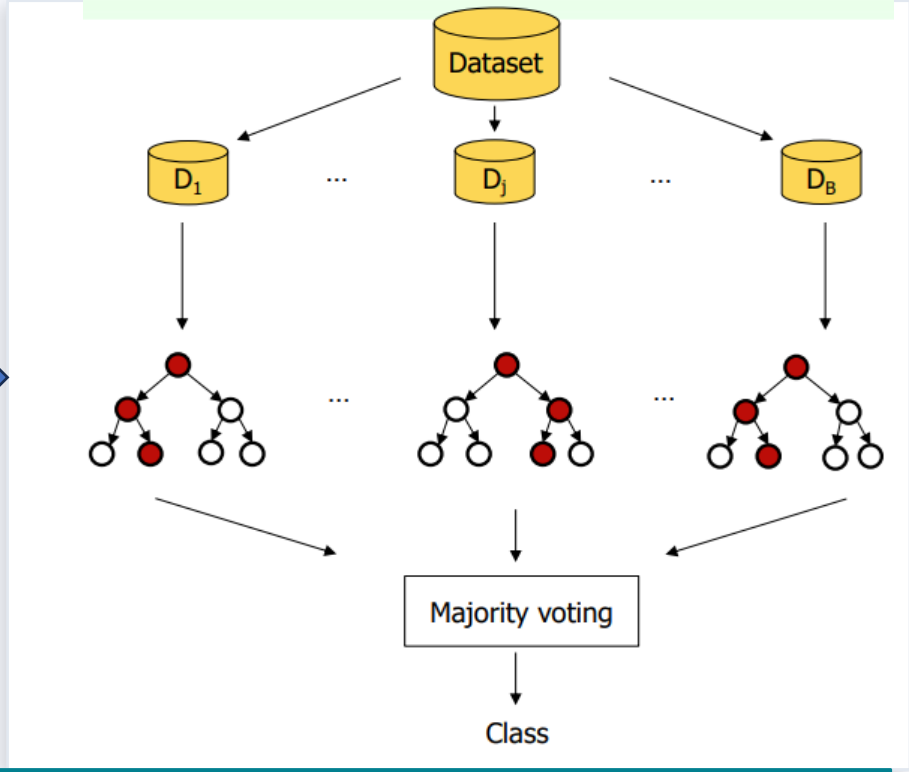
Random Forest

Che cos'è un Random Forest?

Random Forest è una tecnica di classificazione basata su più decision tree (**ensamble learning**) .
Genera **N** alberi , ciascuno addestrato su un subset dei dati di training. Per ogni nodo , le biforcazioni vengono decise valutando l'impurità solo su un sottoinsieme degli attributi.

Funzioni e caratteristiche principali:

- **Algoritmo robusto ad outlier e rumore**
- **Suddivisione dei Dati (Splitting):** Divide il dataset in sotto-gruppi più omogenei rispetto alla classe di appartenenza, come nel decision tree, ma tal decisione si basa solo su un subset di attributi
- **Accuratezza migliore di un singolo Decision Tree**
- **Efficienza**



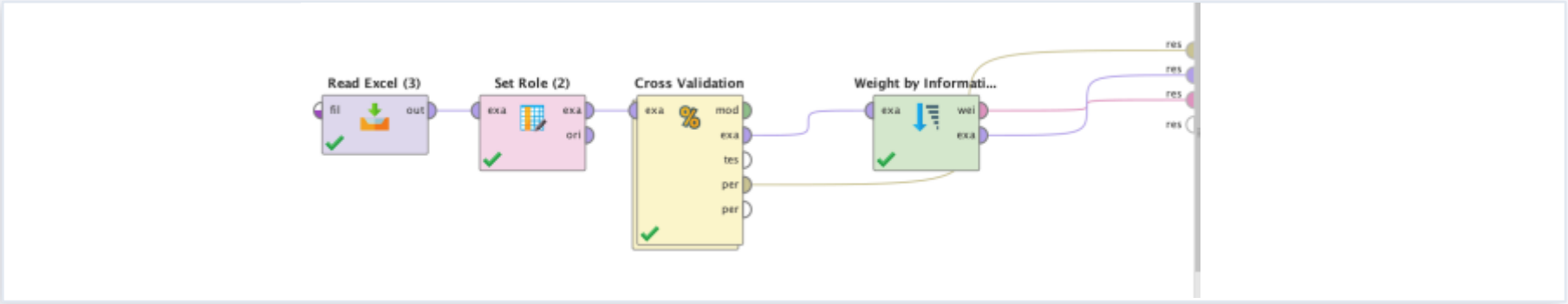
Nel Lab: aggiungere alla pipeline "Read Excel" -> "Set Role" -> "Random Forest"

Interpretare classificazioni con Random Forest

Come interpretare una classificazione fatta usando Random Forest?

La decisione finale è presa combinando le predizioni di molti alberi diversi , usando votazione di maggioranza. Analizzare ogni albero sarebbe un processo lungo e complesso.

Per interpretare il modello, si possono valutare quali siano le **feature più importanti per la classificazione** , **valutando i loro pesi.**



Nel Lab: "weight by information Gain Ratio" è il blocco che permette di valutare la rilevanza degli attributi.

Obiettivo 2

Ottimizzare parametri

Come ottimizzare i parametri di un operatore in Rapid Miner ?

Ottimizzare Parametri in Rapid Miner

Come farlo?

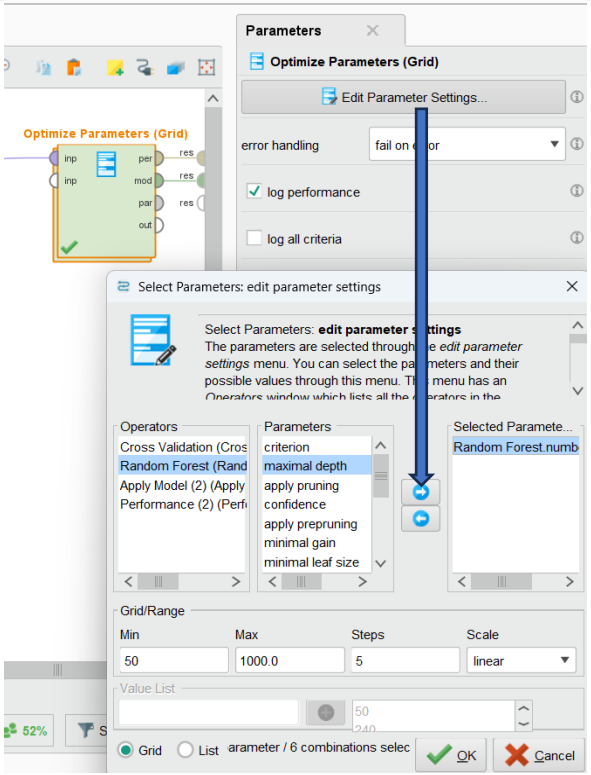
In Rapid Miner è presente operatore **Optimize Parameter(Grid)**. Esso permette di eseguire i modelli al suo interno più volte.

Ad ogni ciclo testa una diversa combinazione di parametri settata dall'utente.

Infine, restituisce una tabella con le perfomance ottenute con ciascun set di parametri modificati.

Processo

- Inserire l'operatore "**Cross Validation**" all'interno di "**Optimize Parameters (Grid)**".
- All'interno della "Cross Validation", strutturare il processo come nel precedente laboratorio:
  - Lato sinistro (Training): Inserire l'algoritmo di machine learning (Es. Random forest).
  - Lato destro (Testing): Inserire l'operatore "Apply Model" collegato all'operatore "Performance".



Nel Lab: dentro l'operatore "Optimize Parameter " in edit parameters settings , si specifica quale parametri modificare e che valori essi assumano

Obiettivo 3

SVM

Support Vector Machine

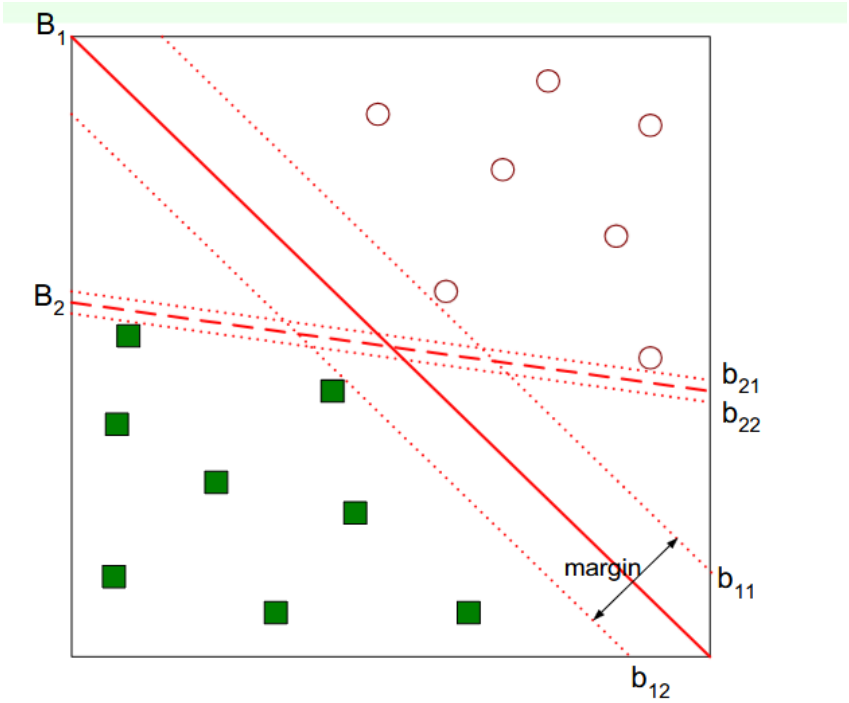
Support Vector Machine

Che cos'è Support Vector Machine (SVM)?

SVM è una tecnica di classificazione basata sul trovare il "**decision boundary**" più adatto a separare i dati.
Tra i possibili "boundary" , sceglie quello che **massimizza** i margini.
In base ai parametri decisi , può apprendere boundary lineari o non lineari . Alcuni dataset potrebbero essere separabili solo da boundary non lineari , ma potrebbero essere più soggetti ad **overfitting** e rumore.

Funzioni e caratteristiche principali:

- Algoritmo robusto ad outlier e rumore
- Richiede tuning dei parametri
- Classificazione rapida
- Poco interpretabile (black box)



Nel Lab: aggiungere alla pipeline "Read Excel" -> "Set Role" -> "Support Vector Machine (libSVM)" con kernel type desiderato

Pipeline SVM

Fase 1 – Preprocessing (Users.xls)

1. Read Excel	2. Nominal To Numerical	3. Set Role	4. Sample
Caricare Users.xls	Seleziono tutti gli attributi tranne response , per convertirli in valori numerici	Response → label	Riduco il dataset di training mantenendo solo il 10% degli esempi

Fase 2 – Applicare il modello e valutazione training error (Users.xls)

5. SVM	6. Apply Model	7. Performance
Applica il modello SVM usando varie configurazioni di kernel	Predice la classe per ogni User	Calcola l'accuratezza sul training set. ≠ prestazioni valutare prestazioni reali (**Data Leakage**)

Nel Lab: nel operatore "**Sample**" impostare parametro "**Sample**" a "**Relative**" e "**Sample Ratio**" a 0.1

Obiettivo 4

Reti Neurali

Come usare Reti Neurali per la classificazione?

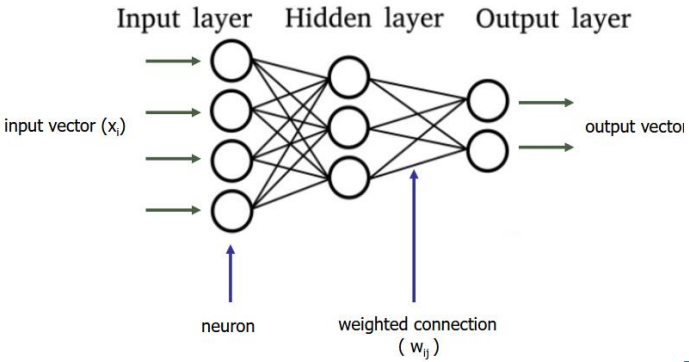
Reti Neurali

Cosa sono le reti neurali?

- Le reti neurali sono modelli di Deep Learning usati per eseguire classificazione .
- Nel laboratorio usiamo reti feed forward .
- Esse sono composte da un *input* layer , un *output* layer e da **N** *hidden layers*.
- Ogni layer è formato da **neuroni** , ogni neurone trasforma i dati ricevuti dal layer precedente.
- Calcola una combinazione lineare dei dati e dei pesi, aggiunge termine di bias ed applica una funzione di attivazione.
- L'obbiettivo del processo di training è imparare i pesi e i bias affinchè l'output sia più vicino possibile alla classe reale.

Funzioni e caratteristiche principali:

- Algoritmo robusto ad outlier e rumore
- Difficile da configurare, richiede di settare molti parametri (n layer, dimensione layer , ecc ..)
- Classificazione rapida
- Scarsamente interpretabile (black box)
- Richiede molti dati per il training
- Ottima accuratezza (se settato bene)



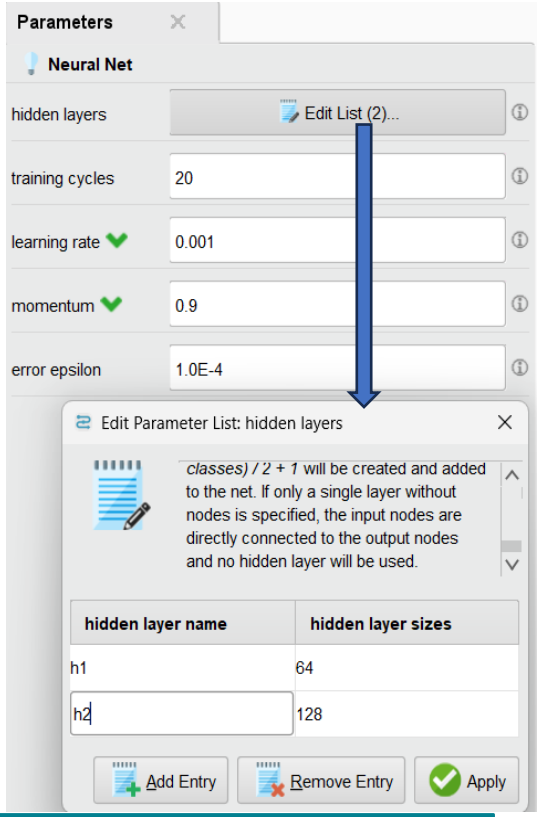
Nel Lab: in rapidminer l'operatore "Neural Net" permette d'imparare un modello usando una rete neurale feed forward

Reti Neurali

Parametri Reti Neurali

Nelle reti FFNN(Feed Forward Neural Networks) bisongna settare vari parametri , quali ad esempio:

- **Numero di layer**: Definisce quanto il modello può comprendere relazioni complesse, gerarchiche, ma potrebbe portare ad overfitting.
- **Dimensione layers** : Indica il numero di neuroni presenti in un layer. Più neuroni ci sono, più dettagli la rete può catturare.
- **Learning rate** : Determina la quanto ad ogni ciclo di apprendimento la rete cambi i parametri. Alto = apprendimento veloce ma potrebbe non trovare l'ottimo, Basso = preciso ma lento.
- **Training cycles** : Indica quante volte la rete elabora l'intero dataset durante il training.
- **Momentum**: Indica quanto il passo attuale viene influenzato da quello precedente. Se l'algoritmo procede nella stessa direzione, il momentum fa accelerare, sennò rallenta le oscillazioni . Aiuta l'algoritmo a non bloccarsi in in minimi locali e velocizza l'apprendimento.



In rapidminer : operatore "Neural Net" in "Hidden layers" > "Edit List", per inserire ed impostare la dimensione di ogni layer.

Obiettivo 5

Overfitting ed Underfitting

Come riconoscerli e combatterli ?

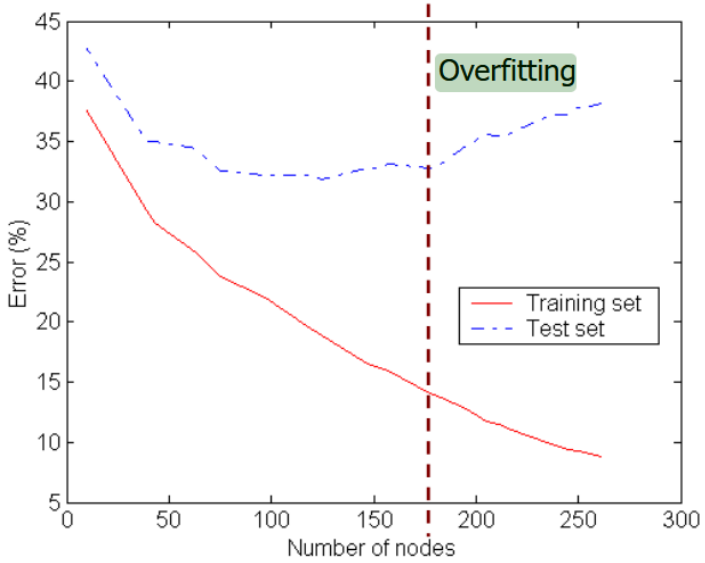
Overfitting ed Underfitting

Overfitting

Durante il processo di training si potrebbe notare un aumento dell'accuratezza sul training set mentre la stessa non migliora sul test set, in questo caso si parla di **Overfitting.**
Il modello si focalizza sui dati del training set e mal si adatta ad essere generalizzato sul test set.

Possibili soluzioni:

- Aumentare i dati per il training
- Ridurre la complessità della rete
- Regolarizzare



Underfitting

Durante il processo di training si potrebbe notare che l'accuratezza rimane bassa sia sul training set che sul test set, in questo caso si parla di **underfitting**
Il modello potrebbe essere troppo semplice e non aver compreso bene le feature presenti nei dati.

Possibili soluzioni:

- Aumentare complessità della rete (n layer o dimensioni)
- Aumentare il numero di epoche
- Data preprocessing

Nel Lab: in rapidminer l'operatore "Neural Net" permette d'imparare un modello usando una rete neurale feed forward

Riepilogo del Lab

① Import dati

② Random Forest

③ Ottimizzazione parametri

④ SVM

⑤ Reti neurali

⑥ Valutazione

Buon laboratorio!