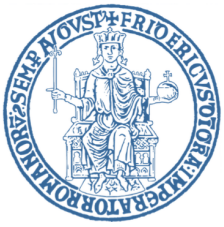
UNIVERSITÁ DEGLI STUDI DI NAPOLI
FEDERICO II



CORSO DI LAUREA IN
INGEGNERIA INFORMATICA

Anno accademico 2014-15

Contest Intelligenza Artificiale

Luca Mocerino
Valentin Popov
Vincenzo Romano

6 luglio 2015

Indice

Capitolo 1

INTRODUZIONE

Nel campo delle scienze informatiche, l'intelligenza artificiale é quella branca di studi che si occupa dello sviluppo di software e macchine intelligenti. Per i maggiori ricercatori del settore, l'intelligenza artificiale consiste nello studio e nello sviluppo di agenti intelligenti, dove per agenti intelligenti si intendono dei sistemi in grado di interagire con l'ambiente circostante e ricavare da questo informazioni utili a massimizzare le probabilitá di successo. L'intelligenza artificiale, quindi, si occupa di sviluppare strategie che permettano a programmi o dispositivi elettronici di ragionare, pianificare, apprendere, percepire, comunicare e manipolare oggetti. Un obiettivo piú a lungo termine é quello di realizzare macchine dotate di un'intelligenza generale (Ipotesi dell'intelligenza artificiale forte, teorizzata per la prima volta dal filosofo statunitense *John Rogers Searle*) in grado di sostituire in tutto e per tutto un cervello umano.

In tale elaborato viene presentato lo studio di un semplice sistema dinamico che descrive un **agente intelligente** capace di effettuare predizioni atmosferiche.

1.1 Presentazione del Contest

Nella seguente sezione saranno indicati i passi eseguiti per l'addestramento e il dimensionamento della nostra rete neurale, che nello specifico é la MultiLayerPerceptron, attraverso l'insieme dei campioni forniti nel dataset

messo a disposizione dai docenti. Nel nostro caso dobbiamo progettare un **Rete Neurale** capace di predire se vi sará presenza di nebbia nei dintorni dell'aeroporto di Parigi con un anticipo di tre ore. Quindi la nostra classificazione andrá realizzata in base a soli due classi **NO** e **YES** (rispettivamente Assenza di nebbia - Presenza di nebbia).

Gli elementi forniti per poter partecipare al contest sono :

- **TrainingSet.arff** : tale file contiene i campioni del dataset per *l'addestramento e la valutazione* del sistema di predizione della presenza o meno di nebbia dopo tre ore . Tali campioni ammontano ad un totale di 92756 :

Classe	**Istanze**
NO	90980
YES	1776

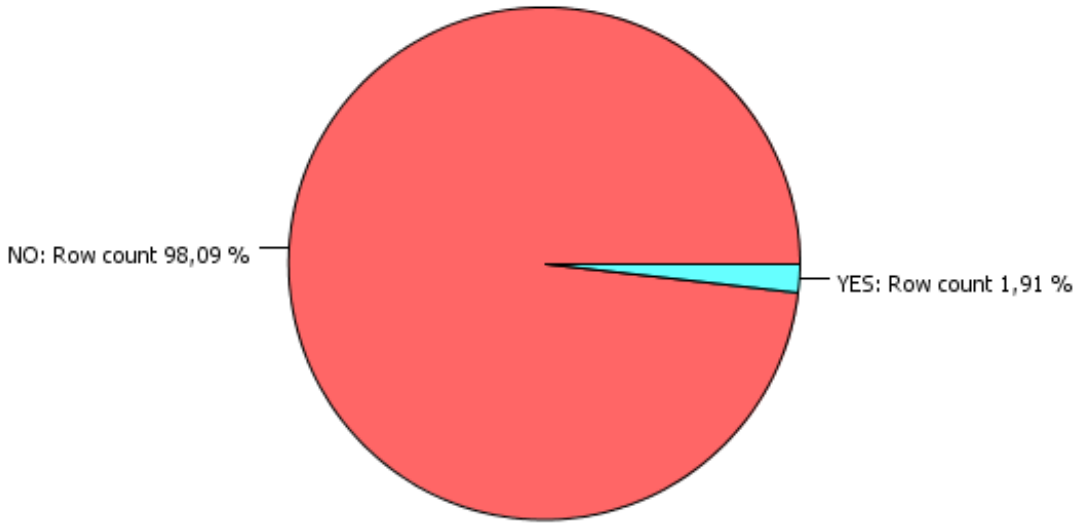


Figura 1.1: Grafico del blocco **Pie Chart**

- **MatricedeiCosti.txt** : Matrice dei costi sulla quale ottimizzare il sistema di predizione della presenza o meno di nebbia. Tale matrice é composta dai seguenti valori :

	NO	**YES**	**Rejected**
NO	0	3	1
YES	10	0	1

- **Meteorological terminology.pdf** : Elenco con la descrizione del significato meteorologico delle feature contenute nel file TrainingSet.arff. In totale le features considerate sono 16 piú la classe **FOG+3** che descrive le previsioni della nebbia nelle successive 3 ore. *(Data la grandezza della tabella raffigurata all'interno di tale documento, ho preferito non inserirla all'interno dell'elaborato)*

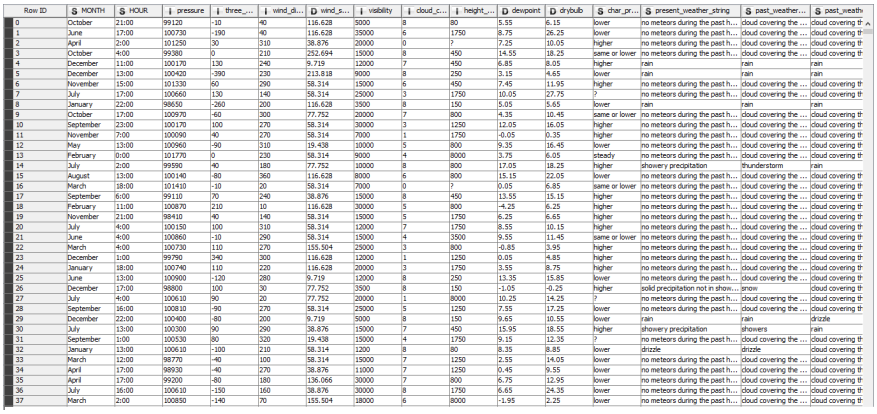
1.2 Obiettivo

L'obiettivo é quello di minimizzare l'errore medio della nostra rete neurale, riducendo al minimo il costo del sistema di classificazione. Tale elaborato é diviso in due parti: la prima é dedicata all'operazione di *Features Selection*, mentre la seconda é dedicata al *Tuning* dei parametri descrittori della rete neurale.

Capitolo 2

DATASET OPERATION

La prima operazione eseguita sul DataSet é quella di acquisire quest'ultimo attraverso l'utilizzo del nodo *ARFF Reader*. Tale nodo necessita come parametro di configurazione il path del file costituente il dataset (TrainingSet.arff); ció che viene restituito in uscita dal nodo é il DS sotto forma tabellare . Tramite il nodo *Color Manager* , selezionando l'opportuna colonna di riferimento (FOG+3) , vengono suddivise le classi di appartenenza NO e YES tramite due colorazioni diverse. A valle dell'esecuzione del Color Manager, possiamo effettuare una **view** della tabella rappresentante il DataSet tramite il blocco *Interactive Table* dalla quale constatiamo che i campioni sono ordinati per classe di appartenenza ; prima tutti i campioni appartenenti alla classe NO , di seguito quelli appartenenti alla classe YES .



Row ID	MONTH	HOUR	pressure	three_...	wind_di...	wind_s...	visibility	cloud_c...	height_...	dewpoint	drybulb	char_pr...	present_weather_string	past_weather...	past_weath
0	October	21:00	99120	-10	40	116.628	5000	8	80	5.55	6.15	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
1	June	17:00	100730	-190	40	116.628	35000	6	1750	8.75	26.25	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
2	April	2:00	101250	30	310	38.876	20000	0	?	7.25	10.05	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
3	October	4:00	99380	0	210	252.694	15000	8	450	14.55	18.25	same or lower	no meteors during the past h...	cloud covering the ...	cloud covering th
4	December	11:00	100170	130	240	9.719	12000	7	450	6.85	8.05	higher	rain	rain	rain
5	December	13:00	100420	-390	230	213.818	9000	8	250	3.15	4.65	lower	rain	rain	rain
6	November	15:00	101330	60	290	58.314	15000	6	450	7.45	11.95	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
7	July	17:00	100660	130	140	58.314	25000	3	1750	10.05	27.75	?	no meteors during the past h...	cloud covering the ...	cloud covering th
8	January	22:00	98650	-260	200	116.628	3500	8	150	5.05	5.65	lower	rain	rain	rain
9	October	17:00	100970	-60	300	77.752	20000	7	800	4.35	10.45	same or lower	no meteors during the past h...	cloud covering the ...	cloud covering th
10	September	23:00	100170	100	270	58.314	30000	3	1250	12.05	16.05	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
11	November	7:00	100090	40	270	58.314	7000	1	1750	-0.05	6.35	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
12	May	13:00	100960	-90	310	19.438	10000	5	800	9.35	16.45	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
13	February	0:00	101770	0	230	58.314	9000	4	8000	3.75	6.05	steady	no meteors during the past h...	cloud covering the ...	cloud covering th
14	July	2:00	99590	40	180	77.752	10000	8	800	17.05	18.25	higher	showery precipitation	thunderstorm	rain
15	August	13:00	100140	-80	360	116.628	8000	6	800	15.15	22.05	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
16	March	18:00	101410	-10	20	58.314	7000	0	?	0.05	6.85	same or lower	no meteors during the past h...	cloud covering the ...	cloud covering th
17	September	6:00	99110	70	240	38.876	15000	8	450	13.55	15.15	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
18	February	11:00	100870	210	10	116.628	30000	5	800	-4.25	6.25	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
19	November	21:00	98410	40	140	58.314	15000	5	1750	6.25	6.65	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
20	July	4:00	100150	100	310	58.314	12000	7	1750	8.55	10.15	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
21	June	4:00	100860	-10	290	58.314	15000	4	3500	9.55	11.45	same or lower	no meteors during the past h...	cloud covering the ...	cloud covering th
22	March	4:00	100730	110	270	155.904	25000	3	800	-0.85	3.95	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
23	December	1:00	99790	340	300	116.628	12000	1	1250	0.05	4.85	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
24	January	18:00	100740	110	220	116.628	20000	3	1750	3.55	8.75	higher	no meteors during the past h...	cloud covering the ...	cloud covering th
25	June	13:00	100900	-120	280	9.719	12000	8	250	13.35	15.85	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
26	December	17:00	98800	100	30	77.752	3500	8	150	-1.05	-0.25	higher	solid precipitation not in show...	snow	cloud covering th
27	July	4:00	100610	90	20	77.752	20000	1	8000	10.25	14.25	?	no meteors during the past h...	cloud covering the ...	cloud covering th
28	September	16:00	100810	-90	270	58.314	25000	5	1250	7.55	17.25	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
29	December	22:00	100400	-80	200	9.719	5000	8	150	9.65	10.55	lower	rain	rain	drizzle
30	July	13:00	100300	90	290	38.876	15000	7	450	15.95	18.55	higher	showery precipitation	showers	rain
31	September	1:00	100530	80	320	19.438	15000	4	1750	9.15	12.35	?	no meteors during the past h...	cloud covering the ...	cloud covering th
32	January	13:00	100610	-100	210	58.314	1200	8	80	8.35	8.85	lower	drizzle	drizzle	cloud covering th
33	March	12:00	98770	-40	100	58.314	15000	7	1250	2.55	14.05	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
34	April	17:00	98930	-40	270	38.876	11000	7	1250	0.45	9.55	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
35	April	17:00	99200	-80	180	136.066	30000	7	800	6.75	12.95	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
36	July	16:00	100610	-150	160	38.876	30000	8	1750	6.65	24.35	lower	no meteors during the past h...	cloud covering the ...	cloud covering th
37	March	2:00	100850	-140	70	155.904	18000	6	8000	-1.95	2.25	lower	no meteors during the past h...	cloud covering the ...	cloud covering th

Figura 2.1: Tabella relativa al **DataSet**

Per evitare che tale ordinamento influisca in maniera negativa sulle prestazioni della rete , andiamo a normalizzare e mescolare il DS tramite i due blocchi *Normalizer* e *Shuffle*. *(non riporto di seguito la nuova tabella , poiché non sono visibili tale modifiche data l'eccessiva quantitá di istanze relative alla classe NO rispetto a quelle relative alla classe YES).*

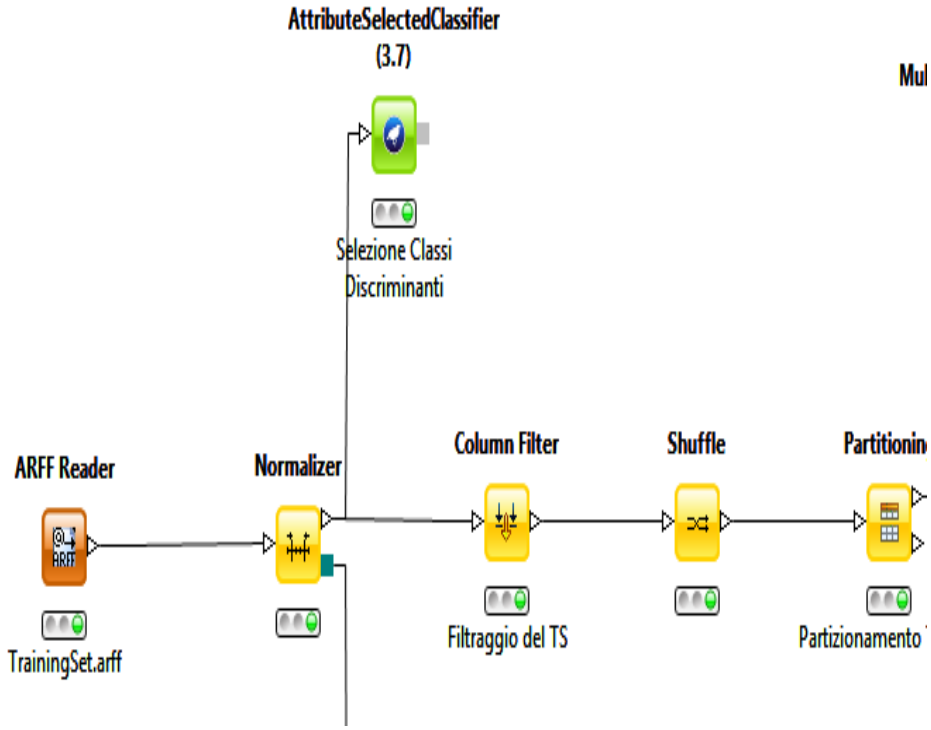


Figura 2.2: Modello **Knime** (Prima Parte)

Capitolo 3

FEATURES SELECTION

Non tutte le features contenute nel DS sono realmente significative per questo introduciamo una fase di *Features Selection* che mira ad individuare quel sottoinsieme di features con cardinalitá minima che consenta di avere un potere discriminante di rilievo.
Benché possa sembrare contraddittorio eliminare delle features dal nostro dataset,tale operazione si rileva di particolare interesse in quanto permette di ridurre la ridondanza di informazioni dovute alle 17 features eliminando eventuali caratteristiche che non possiedono contenuti informativi rilevanti per il contesto .
Conseguentemente diminuisce la complessitá della rete a favore della velocitá di esecuzione operativa. L'operazione di features selection é stata effettuata su tutto il dataset di partenza visto che per questa operazione é meglio prendere in considerazione quante piú informazioni possibile.

3.1 Selezione

La Selezione é stata effettuata attraverso il nodo *AttributeSelectedClassifier* (WEKA). Il nodo mette a disposizione una serie di possibili strategie di ricerca, per ottenere sottogruppi di features significativi e gli Evaluetors al fine di valutarne il potere discriminante.
Come Evaluetor abbiamo utilizzato *Cfs Subset Eval* processo che valuta la capacitá di discriminazione di un sottoinsieme di attributi considerando l'a-

bilitá predittiva di ogni attributo unitamente al loro grado di ridondanza. I sottoinsiemi di attributi aventi un'elevata correlazione con la classe di appartenenza e una bassa intercorrelazione saranno selezionati.
Dopo la scelta dell'Evaluetor mi sono dedicato alla scelta dell'algoritmo di ricerca piú adatto ad ottimizzare il nostro gruppo di features. Gli algoritmi esaminati sono stati:

- **BestFirst** : Questa strategia implementa un'apposita funzione di valutazione ed ha il compito di selezionare, ad ogni passo della ricerca, il successivo nodo da espandere.
  Ad ogni passo, dunque, tra tutti i nodi possibili da espandere l'algoritmo sceglie il nodo con la funzione di valutazione piú bassa.

- **RankSearch** : Questa strategia prevede la valutazione di tutti gli attributi assegnando ad ognuno un punteggio, viene restituito il miglior set di attributi.

- **GreedyStepwise :** Effettua una ricerca Greedy di tipo forward o backward.

Abbiamo quindi adottato l'algoritmo di ricerca Best First.Riportiamo di seguito la risoluzione del blocco *AttributeSelectedClassifier* in merito all'**algoritmo di ricerca BestFirst** :

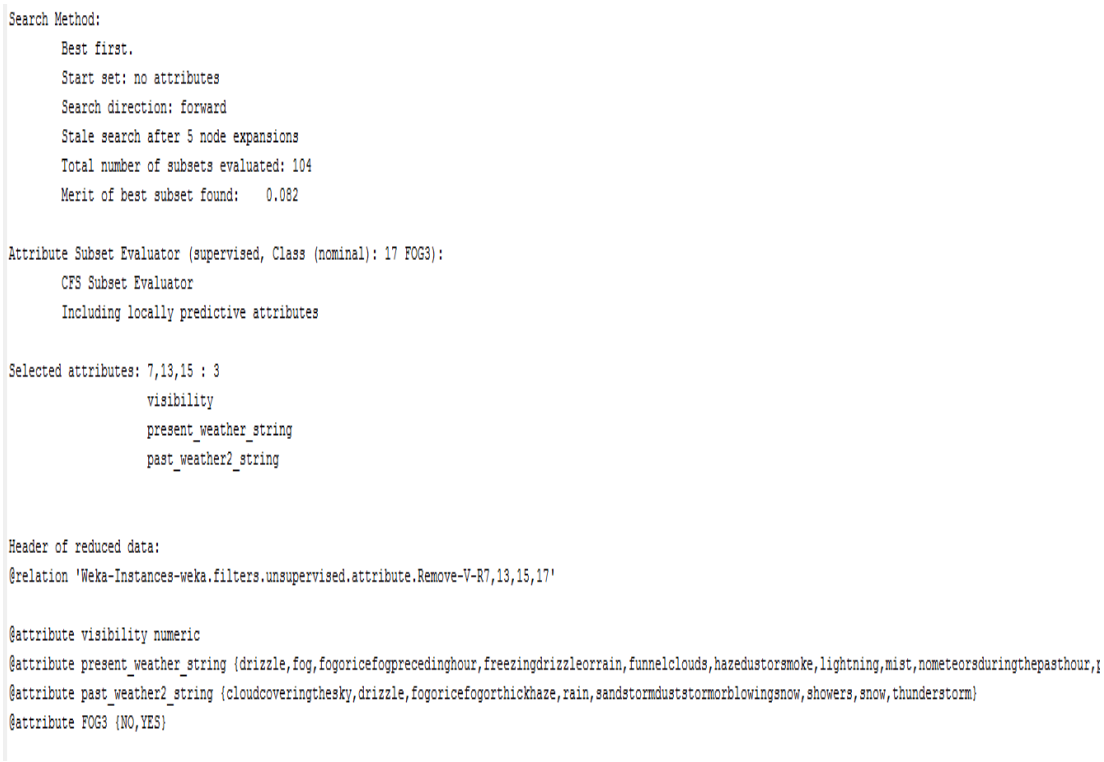


Figura 3.1: Risultati *Algoritmo BestFirst*

Il risultato dell'elaborazione restituisce un set di features di cardinalitá (pari a 3) nettamente inferiore rispetto all'insieme iniziale di partenza (cardinalitá pari a 17).
Dunque l'insieme di features ottenuto per noi rappresenta il sottoinsieme di caratteristiche del DS piú discriminante; utilizzeremo questo risultato per eliminare dal DS l'insieme di colonne che non sono state selezionate in fase di features selection permettendo cosí sia di ridurre la quantitá di dati su cui lavorare, senza degradare in maniera significativa la capacitá di classificazione, e di ridurre il tempo di addestramento della rete neurale.
Quest'azione la implemento configurando in maniera adeguata il nodo *ColumnFilter* nel seguente modo:

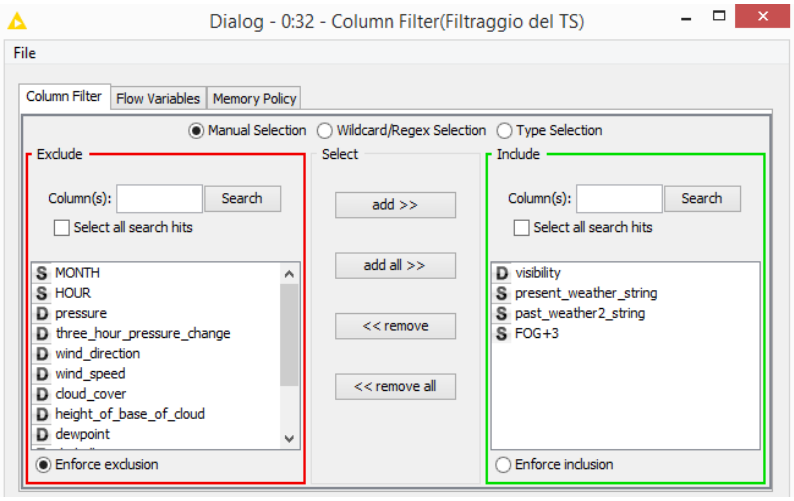
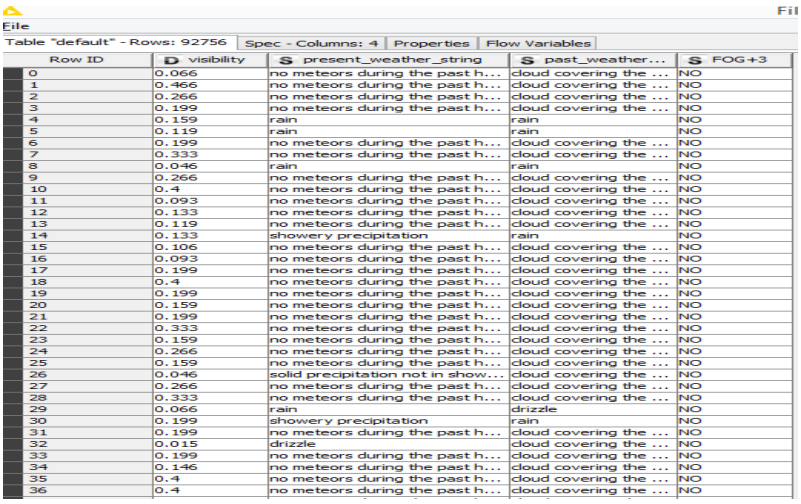


Figura 3.2: Column Filter settings

Tale filtraggio, adesso, permette di selezionare come colonne della tabella relativa al DataSet solo le features piú discriminanti(quelle selezionate dall'algoritmo BestFirst). Pertanto la tabella in uscita dallo shuffle conterrá solo 3+1 featureas, l'1 é relativo alla classe FOG+3:



Row ID	visibility	present_weather_string	past_weather...	FOG+3
0	0.066	no meteors during the past h...	cloud covering the ...	NO
1	0.466	no meteors during the past h...	cloud covering the ...	NO
2	0.266	no meteors during the past h...	cloud covering the ...	NO
3	0.199	no meteors during the past h...	cloud covering the ...	NO
4	0.159	rain	rain	NO
5	0.119	rain	rain	NO
6	0.199	no meteors during the past h...	cloud covering the ...	NO
7	0.333	no meteors during the past h...	cloud covering the ...	NO
8	0.046	rain	rain	NO
9	0.266	no meteors during the past h...	cloud covering the ...	NO
10	0.4	no meteors during the past h...	cloud covering the ...	NO
11	0.093	no meteors during the past h...	cloud covering the ...	NO
12	0.133	no meteors during the past h...	cloud covering the ...	NO
13	0.119	no meteors during the past h...	cloud covering the ...	NO
14	0.133	showery precipitation	rain	NO
15	0.106	no meteors during the past h...	cloud covering the ...	NO
16	0.093	no meteors during the past h...	cloud covering the ...	NO
17	0.199	no meteors during the past h...	cloud covering the ...	NO
18	0.4	no meteors during the past h...	cloud covering the ...	NO
19	0.199	no meteors during the past h...	cloud covering the ...	NO
20	0.159	no meteors during the past h...	cloud covering the ...	NO
21	0.199	no meteors during the past h...	cloud covering the ...	NO
22	0.333	no meteors during the past h...	cloud covering the ...	NO
23	0.159	no meteors during the past h...	cloud covering the ...	NO
24	0.266	no meteors during the past h...	cloud covering the ...	NO
25	0.159	no meteors during the past h...	cloud covering the ...	NO
26	0.046	solid precipitation not in show...	cloud covering the ...	NO
27	0.266	no meteors during the past h...	cloud covering the ...	NO
28	0.333	no meteors during the past h...	cloud covering the ...	NO
29	0.066	rain	drizzle	NO
30	0.199	showery precipitation	rain	NO
31	0.199	no meteors during the past h...	cloud covering the ...	NO
32	0.015	drizzle	cloud covering the ...	NO
33	0.199	no meteors during the past h...	cloud covering the ...	NO
34	0.146	no meteors during the past h...	cloud covering the ...	NO
35	0.4	no meteors during the past h...	cloud covering the ...	NO
36	0.4	no meteors during the past h...	cloud covering the ...	NO

Figura 3.3: Nuova tabella del *DS*

Capitolo 4

FASE DI ADDESTRAMENTO MLP

MultiLayerPerceptron é una rete a piú strati o layers, per la precisione 2 layers: le componenti di un vettore di input X di cardinalitá N entrano nel layer di ingresso, che non effutta nessun tipo di elaborazione e si propagano, attraverso gli N neuroni che compongono questo strato, direttamente in quello successivo senza subire alcuna modifica (non é, a rigore, un layer in senso stretto in quanto i neuroni non effettuano alcuna elaborazione); esiste almeno un layer nascosto (*Hidden Layer*) che non vedra' ne gli ingressi ne le uscite della rete; infine, il layer di uscita emette un vettore Y di cardinalitá M che costituisce l'output della rete.Ciascun neurone, esclusi quelli del primo layer, riceve un numero di ingressi pari al numero dei neuroni dello strato precedente, piú la Bias Unit. In seguito si effettua il prodotto scalare tra tali ingressi e il proprio vettore dei pesi (rappresentativi della memoria locale del singolo percettrone) cui viene sommato il peso associato alla Bias Unit; calcolata questa combinazione, ad essa viene poi applicata la funzione di attivazione producendo l'uscita effettiva del neurone che si propaga nella rete esclusivamente in avanti.

Nel nostro caso la fase di addestramento del percettrone multilivello viene implementata attraverso il seguente insieme di blocchi in Knime:

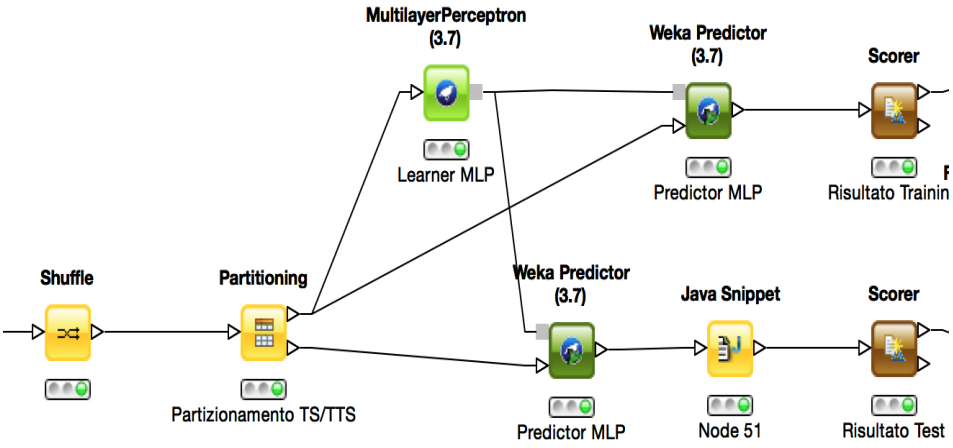


Figura 4.1: Modello **Knime** (Seconda Parte)

Vediamo in dettaglio i seguenti blocchi:

- **Partitioning :** settato nel seguente modo:

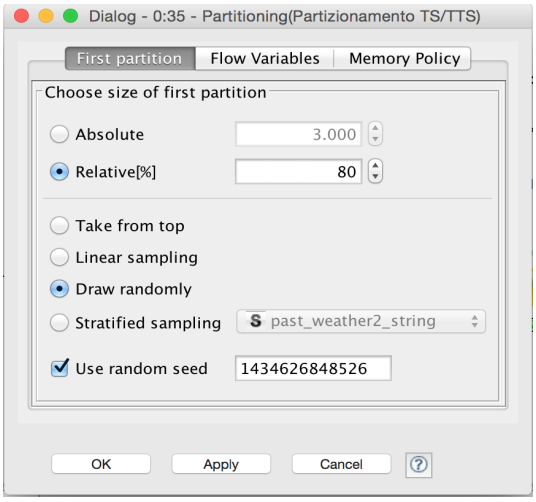


Figura 4.2: Setting del blocco *Partioning*

Tale blocco esegue una suddivisione del DS filtrato in *Test Set(TS)* e *TrainingTestSet(TTS)*. La prima partizione, costituita dall' 80% del DS, e utilizzata per addestrare la rete; la seconda partizione , costituita dal restante 20% del DS , viene utilizzata per verificare la bonta del classificatore in termini di errore percentuale.

- **MultilayerPerceptron :** anche detto *LearnerMLP* ,tale blocco é dedito all'addestramento del percettrone . Per la spiegazione di tale blocco dedicheremo il subsection successiva.

- **WekaPredictor :** tale blocco riceve in ingresso l'uscita del blocco MLP e Partitioning sviluppando la classificazione delle varie istanze. In particolare abbiamo due di questi blocchi , uno per classificare le istanze in base al TS mentre l'altro classifica le istanze in base al TTS.

- **Scorer :** ricevuta in ingresso l'uscita del WekaPredictor, ne fornisce una rappresentazione sottoforma tabellare.

4.1 Configurazione MLP Learner

Nel setting del blocco *MLP Learner* andiamo a settare i seguenti parametri:

- **hiddenlayer :** corrisponde al numero di *strati nascosti* che formano il mio MLP;
- **laerningrate :** paramento minore di uno solitamente settato intorno a 0.5 (come nei casi seguenti);
- **momentum :** parametro che é stato settato a 0.5 per tutti i test seguenti;

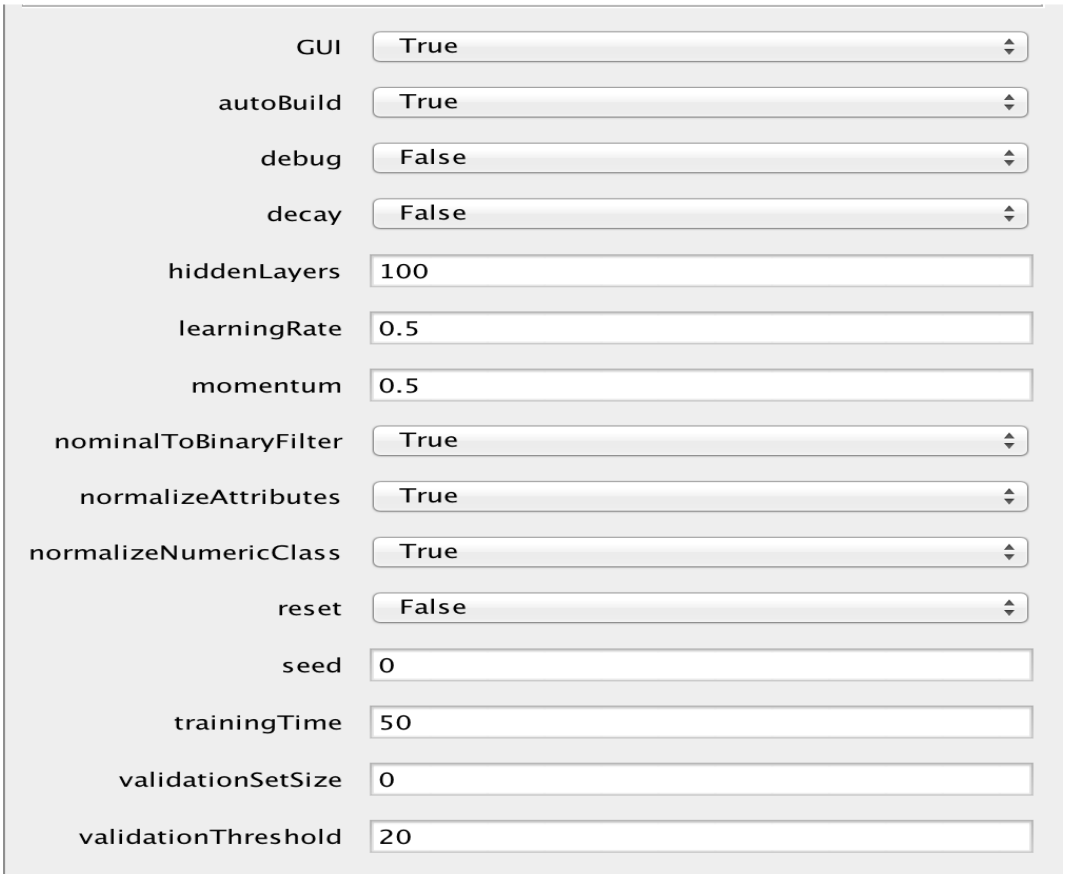


Figura 4.3: Esempio di settaggio dei parametri del *MLP Learner*

4.2 Prove Addestramento

Ora, andremo ad eseguire vari addestramenti del MLP per verificare quale sia il migliore.

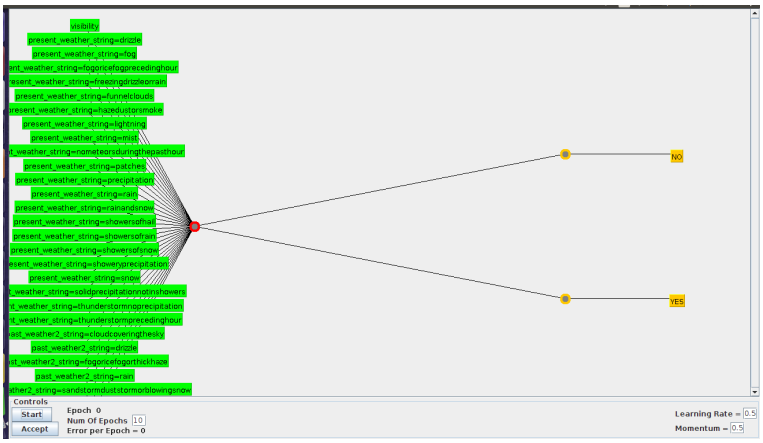
4.2.1 Problemi di Overfitting e Overtrainig

Fra i problemi del percettrone a due livelli abbiamo

- **Minimi locali:**
  che si risolve ricorrendo alla tecnica di back propagation.
- **Flat spots**:
  una errata inizializzazione dei pesi,che porta nella zona dove la derivata dell'errore e praticamente nulla. Si risolve inizializzando il vettore dei pesi al valore $+/-\sqrt{n}$.
- **Overfiting**;
- **Overtainig**;

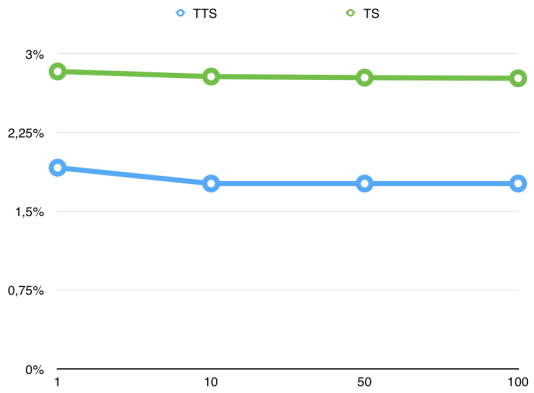
Ł'effetto del overfitting e overtrainig e' lo stesso: la rete aderisce troppo al training set e quindi perde in generalizzazione (si specializza troppo), cioe' non classifica bene gli esempi del test set. Pero' le cause sono diverse:
- *causa del overfitting*: troppi neuroni nello strato hidden.
- *causa del overtraining*: addestramento eccessivo della rete (troppe epoche).
Di seguito vengono riportate le prestazione della rete in varie configurazioni:

Cofigurazione 1: **1 neurone** nel livello hidden:

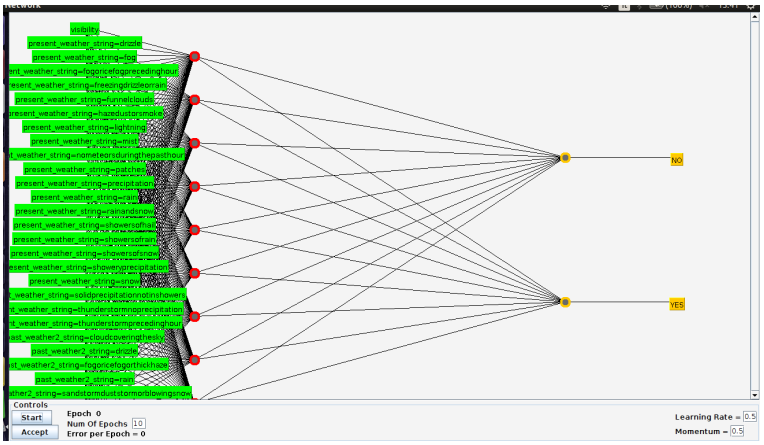


(Errore/Epoche)(n°neuroni=1)

	1	10	50	100
TTS	1,914%	1,763%	1,763%	1,763%
TS	2,83%	2,781%	2,771%	2,765%

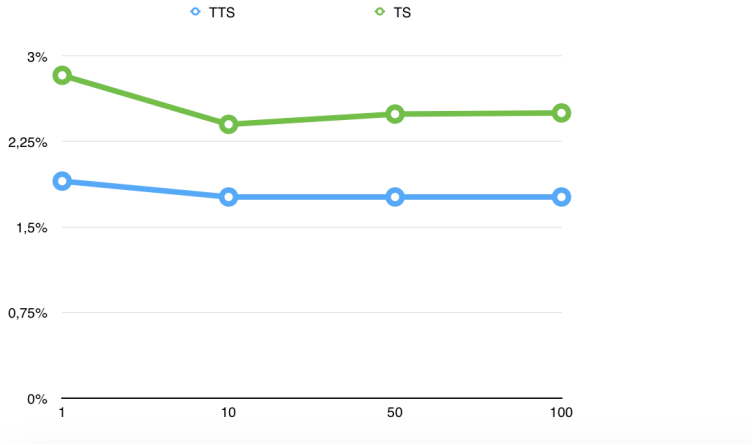


Cofigurazione 2: **2*4+1=9 neuroni** nel livello hidden:

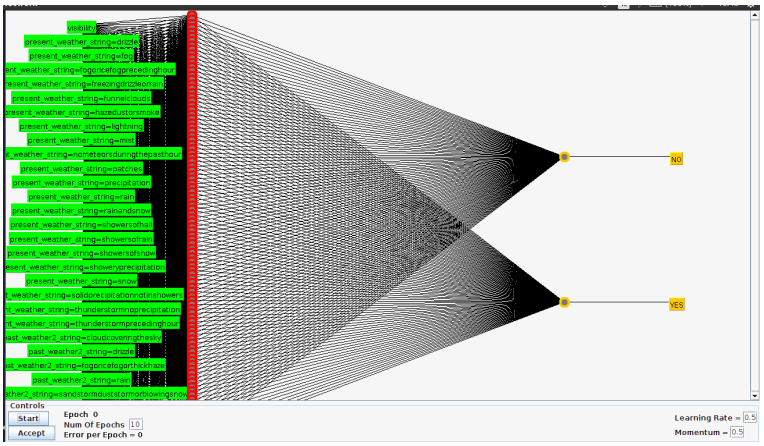


(Errore/Epoche)(n°neuroni=9)

	1	10	50	100
TTS	1,902%	1,763%	1,763%	1,763%
TS	2,83%	2,4%	2,49%	2,5%

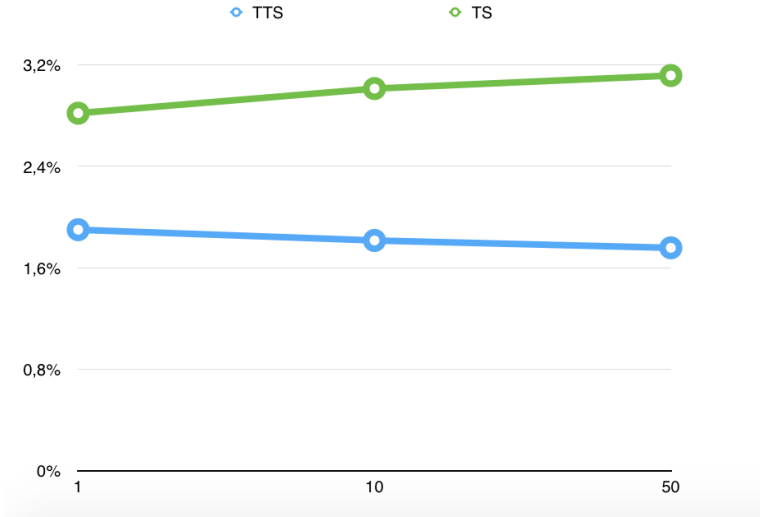


Cofigurazione 3: **100 neuroni** nel livello hidden:



(Errore/Epoche)(n°neuroni=100)

	1	10	50
TTS	1,9%	1,815%	1,757%
TS	2,819%	3,013%	3,116%



Capitolo 5

ANALISI DEI DATI OTTENUTI

Il riusultato migliore ottenuto e' congruente con il **Teorema di Kolmogorov**: *Considerata una qualunque funzione continua ad n ingressi e m uscite, che assumono valore tra 0 e 1 (opportunamente normalizzate), puo' essere esattamente implementata da una RN FF a 3 stati con* $(2n + 1)$ *neuroni nello strato hidden.* Dalle performance ottenute possiamo dedurre la configurazione migliore e' la seguente:

- 9 neuroni nel hidden layer;
- 10 epoche di addestramento;

Capitolo 6

ANALISI DEI COSTI

Per la *stima del costo* di classificazione abbiamo utilizzato il blocco Knime *Cost* e un *File Reader* che ci ha consentito di caricare la matrice dei costi.

Ne risulta che il costo di classificazione é pari a **0.1**.