Consiglio Nazionale delle Ricerche

Data Mining in Ambiente Java
un Caso di Studio sul
Data Warehouse Piano Telematico Calabria

Fosca Giannotti, Giuseppe Manco
Sabrina Tardelli, Loredana Versienti

Rapporto
CNUCE-B4-2000-031

Pisa

Data Mining in Ambiente Java
un Caso di Studio sul
Data Warehouse Piano Telematico Calabria

Fosca Giannotti, Giuseppe Manco, Sabrina Tardelli, Loredana Versienti

Abstract. Questo rapporto tecnico descrive una serie di esperimenti sistematici di analisi dei dati del Data Warehouse Piano Telematico Calabria mediante tecniche di mining. Si mostra una panoramica dell'ambiente di sviluppo utilizzato, contenente i package che implementano la maggior parte degli algoritmi del processo *Knowledge Discovery in Databases* (libreria Weka). Sono presentate esperienze di pre-processing ed analisi sul datamart contenente informazioni socio-economiche del Sistema Calabria. Il lavoro e' stato svolto utilizzando e sviluppando tecniche esplorative e tecniche predittive. L'obiettivo, nel primo caso, e' l'estrazione di conoscenza non ovvia dalle sorgenti informative; nel secondo caso, la realizzazione di un meta-classificatore delle imprese calabresi capace di predire la continuità o meno delle attività delle imprese rispetto alla tipologia del comune in cui sono ubicate.

1. INTRODUZIONE

La crescita sempre più massiccia della quantità di informazioni disponibili e la loro aumentata "raggiungibilità", ha portato allo sviluppo di metodologie e strumenti che permettono di elaborare i dati originari e ricavarne informazioni non ovvie e di cruciale importanza per l'utilizzatore finale. Attualmente, milioni di database vengono utilizzati nelle attività commerciali, bancarie, scientifiche ed in molte altre applicazioni, il loro numero cresce rapidamente sia a causa della facilità di reperimento e memorizzazione dei dati, sia a causa della crescita di sistemi di gestione di base di dati sempre più potenti ed affidabili.

La disponibilita' di enormi quantita' di dati ha fatto nascere l'interesse per una analisi dei dati stessi, al fine di estrarre informazioni. L'estrazione di conoscenza da grosse quantita' di dati (Data Mining, DM) rappresenta il principale obiettivo di un insieme di tecniche e di strumenti che fanno parte di un processo noto in

letteratura con il nome di *Knowledge Discovery in Databases* (KDD), [PT99], [M96]. In letteratura esistono differenti definizioni di KDD e di DM che dimostrano la continua evoluzione di tali soggetti. Di seguito citiamo una delle prime definizioni che mette in evidenza la complessita' del processo di KDD [FPS96]:

> *" Knowledge Discovery in Databases è il processo non banale di identificazione di modelli validi, nuovi, potenzialmente utili e facilmente comprensibili, che caratterizzino i dati"*

Un problema aperto nell'ambito della ricerca, è quello sulla mancanza di un framework generale in cui sia possibile esprimere l'intero proceso di KDD, [CHY96]. La ricerca ha infatti prodotto un buon numero di tool ad hoc per le varie fasi del processo che hanno raggiunto degli ottimi livelli di efficienza e scalabilità, ma che tuttavia, non sono progettati per interagire nel contesto di un ambiente di supporto all'intero processo. In questo contesto e' stato sviluppato e reso disponibile un sistema di supporto all'intero processo KDD che propone l'utilizzo di strumenti in maniera uniforme ed integrata: WEKA. Il sistema WEKA è stato progettato per integrare un insieme di tecniche e schemi di machine learning sotto una interfaccia uniforme, con l'obiettivo di ottenere una omogenea applicazione sui dati; il software è scritto interamente in Java per facilitare l'accessibilità degli strumenti di Data Mining in modo indipendente dalla piattaforma del computer. WEKA fornisce quindi una collezione di algoritmi e una struttura di interfacciamento con i quali è possibile seguire il processo di estrazione di conoscenza in tutte le sue fasi.

2. LA LIBRERIA WEKA

WEKA, [GH94], Waikato Environment for Knowledge Analysis, è un ambiente sviluppato all'Università di Waikato in Nuova Zelanda nel quale è possibile seguire il processo di estrazione di conoscenza in tutte le sue fasi:

a. **Data understanding**, tramite strumenti statistici per lo studio delle caratteristiche dei dati;

b. **Data pre-preparation**, attraverso un vasto insieme di strumenti che permettono di manipolare i dati sia a livello di tuple che di valori (es: discretizzatore, AttributeSelection, etc.);
c. **Data Mining**, con le implementazioni di molti algoritmi per la predizione e l'esplorazione dei dati (es: C4.5, Apriori, etc.);
d. **Evaluation,** con le implementazioni di alcuni meta learner e tecniche di valutazione, di modelli e pattern estratti (es: Boosting, Cross-Validation, etc.).

Il software WEKA è scritto interamente in JAVA, un linguaggio di programmazione Object Oriented largamente diffuso sulla maggior parte delle piattaforme di computer, e WEKA è stato testato sotto i sistemi operativi Linux, Windows e Macintosh, [FW00].

L'uso di JAVA comporta vari vantaggi:

- Fornisce un'interfaccia uniforme ai differenti algoritmi di learning e ai metodi per la manipolazione dei dati e dei risultati.
- Le applicazioni di WEKA possono girare su ogni computer con un browser WWW; questo permette di applicare le tecniche di machines learning sui propri dataset indipendentemente dalla piattaforma del computer.

L'uso del linguaggio JAVA consente inoltre di utilizzare WEKA a diversi livelli. Le varie implementazioni degli algoritmi possono essere applicate semplicemente a linea di comando: si può pre-processare il dataset, passarlo ad uno schema di learning e analizzare il risultante classificatore e le sue prestazioni, tutto ciò senza scrivere una linea di codice. Un altro livello d'uso di WEKA è richiamare le sue classi dal proprio codice JAVA. Cio' implica che:

- Si possono sviluppare applicazioni ad hoc;
- Gli algoritmi sono facilmente estendibili;
- Si possono di costruire interfacce grafiche al Data Mining.

Ad esempio è possibile utilizzare tutte le funzionalità di WEKA attraverso una interfaccia grafica con la quale si possono specificare le varie fasi del processo di estrazione di conoscenza.

Il prezzo da pagare per utilizzare l'ambiente WEKA sta nel particolare formato in cui devono essere espressi i dati, WEKA infatti utilizza un formato denominato ARFF. Il formato ARFF è una variazione del CSV (Comma Separated Values), nel quale i valori di una tupla di una certa tabella sono separati da una virgola, in più il formato ARFF prevede un'intestazione nella quale sia dichiarato il nome e il tipo degli attributi, in modo da fornire agli algoritmi le informazioni per poter utilizzare i dati. Tale formato prevede che il dataset inizi con la dichiarazione del suo nome preceduto dal tag @relation:

@relation nome relazione

Segue la lista di tutti gli attributi data in questa forma:

@attribute nome-attributo tipo

E' possibile specificare tre soli tipi per gli attributi:

- **numeric** per gli attributi che assumono valori numerici;
- **nominal** per gli attributi che assumono uno tra una specifica lista di valori;
- **string** per gli attributi che vengono utilizzati per memorizzare commenti. Sono attributi che non vengono usati dagli schemi di learning di WEKA; possono essere utilizzati ad esempio per memorizzare un identificatore per ogni istanza del dataset.

Nel caso di *numeric* la dichiarazione è banale ed assume la forma:

@attribute nome numeric

mentre nel caso di *nominal* la dichiarazione è appena più complessa ed il nome dell'attributo deve essere seguito dalla lista dei possibili valori.

@attribute nome {valore1, valore2, valore3 ...}

All'elenco di attributi segue la lista di tutte le istanze preceduta dal tag: @data. Le istanze sono elencate nel formato comma-separated, e il "?" rappresenta un valore perso, [FW00]. La trasformazione dei dati in questo formato è in parte immediata in quanto molti dei moderni sistemi per la gestione dei dati forniscono la possibilità di salvare i dati nel formato CSV introdotto in precedenza; una volta salvati i dati in questo formato bisogna aggiungere le intestazioni come descritto sopra, ciò può comportare un lavoro tedioso per il quale WEKA non fornisce alcuno strumento. Le tabelle 2.1 e 2.2 mostrano un esempio di dati in formato tabellare e la loro corrispondenza nel formato ARFF :

Tempo	Temperatura	Umidita'	Vento	Play Golf
Sole	85	80	Debole	No
Sole	80	90	Forte	No
Coperto	83	86	Forte	Si'
Pioggia	70	96	Debole	Si'
Coperto	83	70	Assente	Si'

Tabella 2.1: Dataset *Golf* in formato tabellare

```
@relation PlayGolf

@attribute Tempo {Sole, Coperto, Pioggia}
@attribute Temperatura numeric
@attribute Umidità  numeric
@attribute Vento{Forte, Debole, Moderato, Assente}
@attribute PlayGolf {Sì, No}

@data

Sole,85,80,Debole,No
Sole,80,90,Forte,No
Coperto,83,86,Forte,Sì
Pioggia,70,96,Debole,Sì
Coperto,83,70,Assente,Sì
```

Tabella 2.2: Dataset *Golf* in formato ARFF

Il formato ARFF può rappresentare un grande limite all'uso di WEKA: se tra i propri dati sono presenti attributi che assumono un numero elevato di valori distinti ad esempio dell'ordine di un migliaio, dichiarare tale attributo diventa proibitivo perché richiederebbe di elencare consecutivamente un migliaio di valori interposti da una virgola.

Come già accennato, WEKA ha la particolarità di poter essere utilizzata a vari livelli, [GH94]:

- E' possibile utilizzare la libreria JAVA che accompagna l'ambiente, nelle proprie applicazioni, in modo da eseguire la fase di estrazione di conoscenza dai dati utilizzando semplicemente le classi fornite da WEKA, concentrandosi sugli aspetti caratteristici delle proprie applicazioni. Di seguito riportiamo un esempio di utilizzo della classe APRIORI nel quale si estraggono da un dataset

chiamato "dati" tutte le regole di associazione con supporto minimo 0.8 e confidenza minima 0.6:

```
apriori = new Apriori();
apriori.setMinSupport(0.8);
apriori.setMinConfidence(0.6);
apriori. buildAssociations(dati);
System.out.println(apriori.toString());
```

Il metodo *toString()* restituisce una stringa contenente una rappresentazione testuale delle regole estratte.

- E' possibile utilizzare a linea di comando i vari algoritmi di WEKA per estrarre conoscenza dai propri insiemi di dati. Il comando base per usare WEKA a questo livello e' *java* che chiama la VM a cui da' le istruzioni per eseguire l'algoritmo. Diamo un esempio di selezione degli attributi 1,2 dal dataset Golf

  java weka.filters.AttributeFilter –R 1,2 –i golf.arff

- E' possibile utilizzare tutte le funzionalità dell'ambiente attraverso una interfaccia grafica.

3. LE SORGENTI INFORMATIVE

Gli esperimenti e le analisi sono state effettuate sulle sorgenti informative contenute nel *Data Warehouse del Piano Telematico Calabria* (DW_PTC). Tale DW_PTC e' formato da un congruo numero di basi di dati, alcune delle quali sono "stabili" e sufficientemente documentate, altre sono in fase progettazione o ristrutturazione e non ancora disponibili. La base di dati contiene varie fonti di informazioni riguardanti svariati aspetti della vita calabrese; ad esempio, sono memorizzate informazioni sul turismo, economia, salute, istruzione, etc.

Le fonti informative che danno vita al nucleo costitutivo del DW-PTC, che al momento possono essere considerate "stabili" sono le seguenti:

1. Sorgente informativa ERICA
2. Sorgente informativa OSCAR
3. Sorgente informativa STUDENTI
4. Sorgente informativa TURISMO

Diamo una presentazione delle BD: ERICA, OSCAR fonti delle nostre analisi.

La base di dati relazionale ERICA, realizzata in ambiente ORACLE, contiene gli Elenchi Registro Imprese Calabresi (ERICA). Offre informazioni sulla natura e sullo stato delle imprese ubicate nel territorio Calabrese, ad esempio, natura giuridica, attività economica primaria, stato dell'Impresa, etc. Le informazioni provengono dal Registro delle Imprese Italiane gestito dalle Camere di Commercio Italiane e sono fornite periodicamente da Cerved S.P.A. Diamo lo schema della BD ERICA.

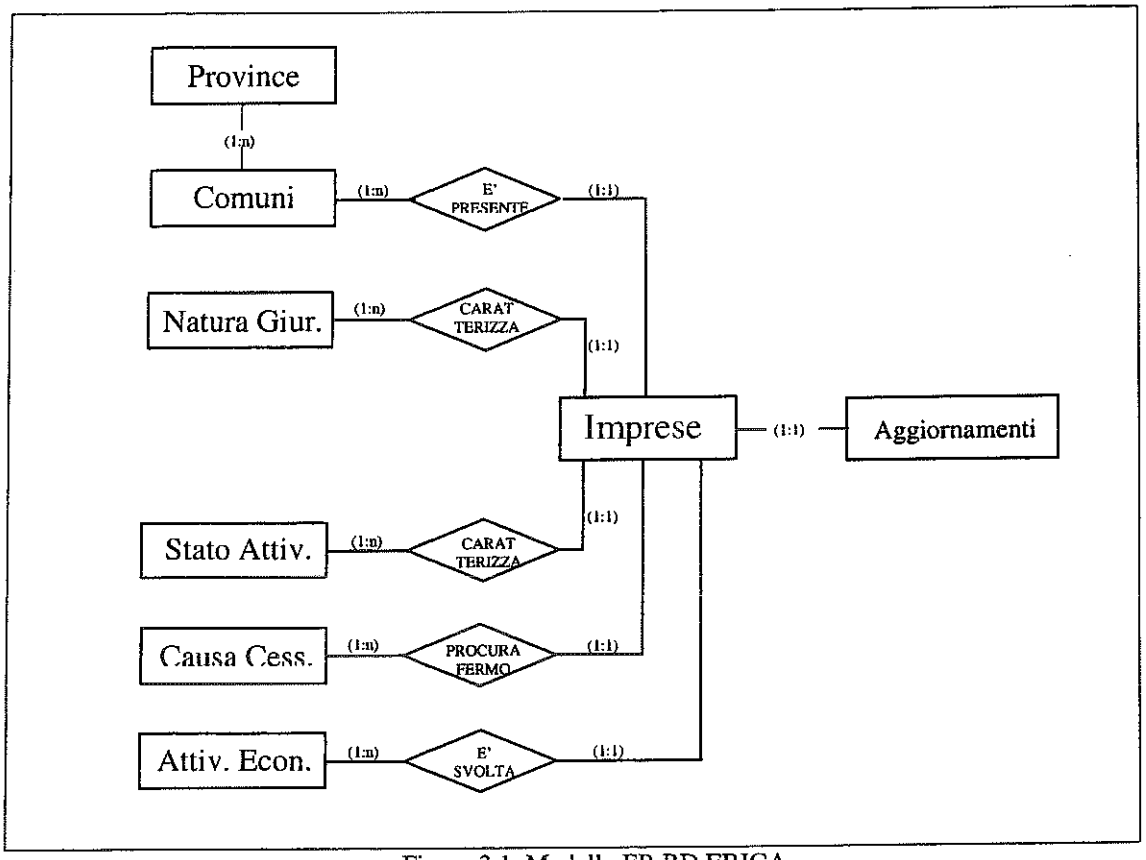


Figura 3.1: Modello ER BD ERICA

La base di dati OSCAR è stata realizzata dall'azione progettuale RICERCA in ambiente ORACLE, contiene i valori di un insieme di indicatori di tipo socio-economico, relativi ad alcuni comuni calabresi (circa 40), alle province calabresi, e al territorio calabrese nel suo complesso. In particolare, OSCAR costituisce un'importante sorgente di informazioni sulle risorse del Sistema Calabria.

I dati in essa contenuti sono classificabili come segue:

- Demografici: popolazione residente, numero di famiglie, etc.;
- Struttura produttiva: consumi di energia elettrica, addetti alle unità locali non agricole, etc;

- Territoriali: distanza di un determinato comune dall'aeroporto o distanza dal capoluogo;
- Condizioni socio-economiche: numero di contribuenti IRPEF, depositi delle aziende di credito, autovetture circolanti, utenze telefoniche privati, etc;
- Dotazione dei servizi: numero di posti letto in istituti pubblici di cura, aule scuola media inferiore, etc;
- Potenziali fruitori della domanda telematica: agenzie marittime, agenti e agenzie di affari, etc;
- Indicatore: connessione alle reti, dimensione demografica, etc;

I suddetti dati si riferiscono al periodo 1991-1998 ma sono, allo stato attuale, largamente incompleti, sono stati ricavati dai dati ufficiali del censimento 1991 o da altre fonti a carattere nazionale (ISTAT, CERVED, ministeri, etc). Diamo lo schema della base di dati OSCAR.

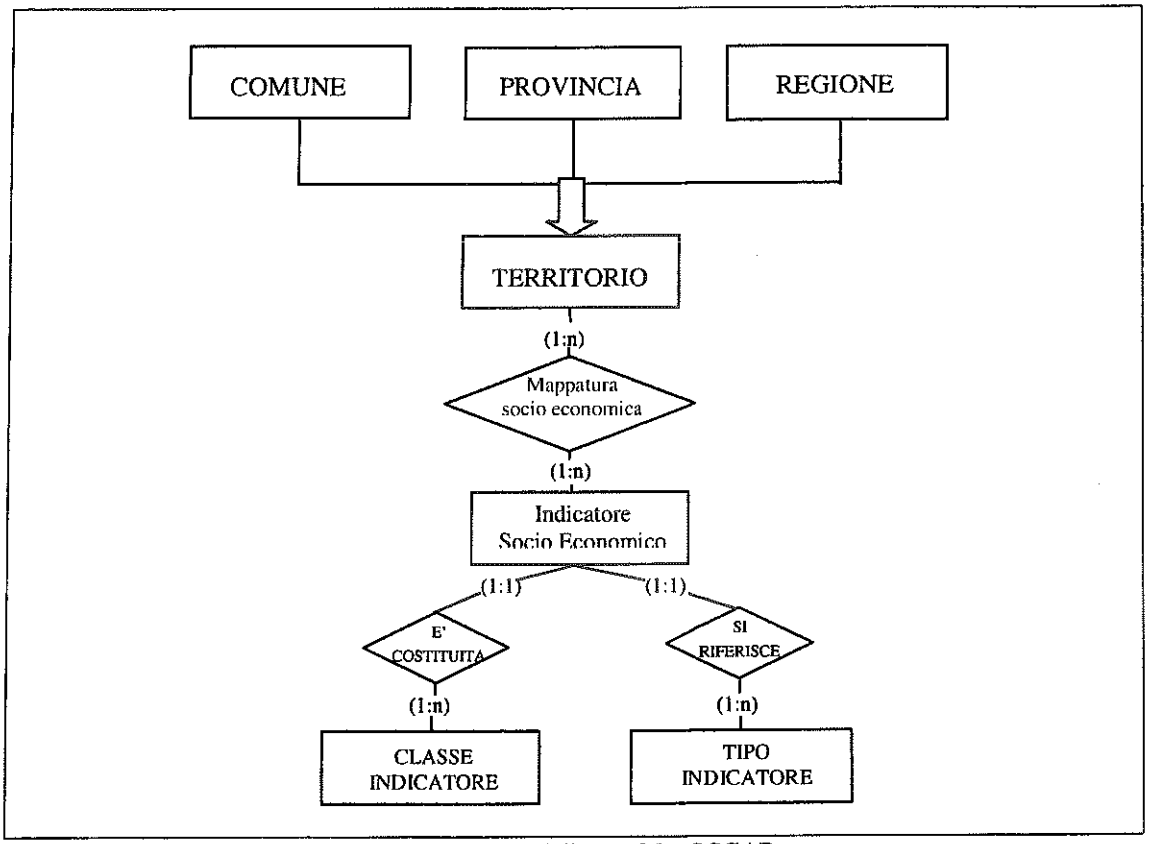


Figura 3.2: Modello ER BD OSCAR

4. DATA PRE-PROCESSING

In questa sezione descriviamo il lavoro svolto per preparare alla fase di Data Mining i dati strutturati delle sorgenti che costituiscono il data warehouse, scegliendo in un primo momento quali dati dovrebbero formare il datamart e riducendo in un secondo tempo le dimensioni degli attributi e i range dei loro valori (discretizzando i valori continui), in modo da ottenere una codificazione dei dati nella rappresentazione appropriata per i particolari algoritmi di Data Mining che abbiamo utilizzato.

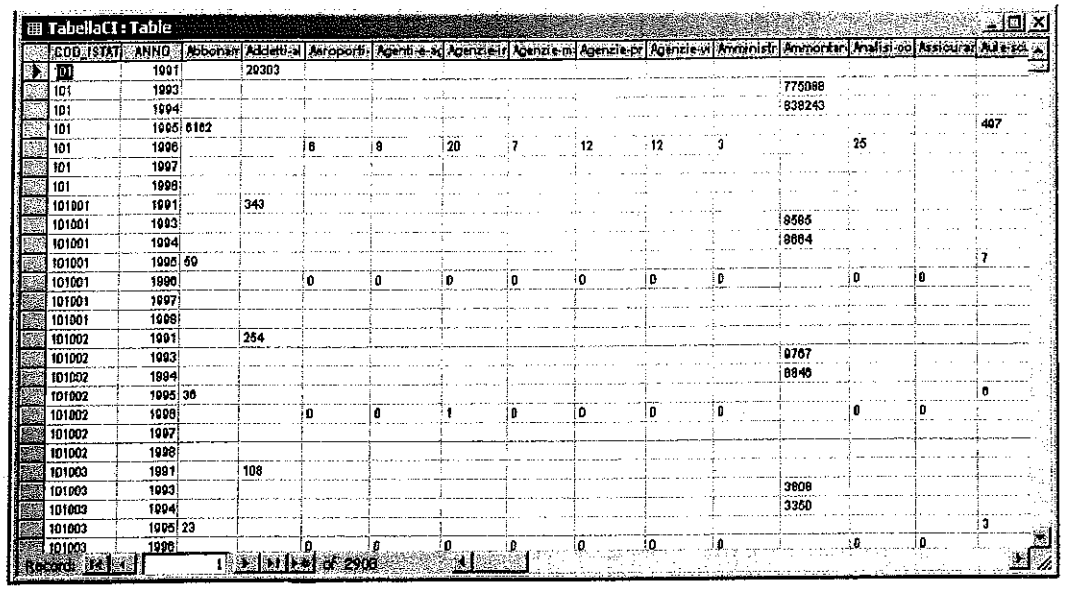
Dalle sorgenti informative OSCAR e ERICA sono stati individuati due insiemi di dati a cui corrispondono due datamart di potenziale interesse per applicazioni di data mining. I datamart individuati sono i seguenti:

- Il datamart relativo al territorio, che fondamentalmente riguarda la sorgente OSCAR a cui abbiamo apportato le necessarie trasformazioni. Questo datamart sarà utilizzato nella fase di mining per effettuare diverse analisi dei legami e cluster analysis su demografia, struttura produttiva, condizioni economiche e servizi.
- Il datamart relativo a territorio e imprese che riguarda lo schema integrato delle basi di dati OSCAR ed ERICA. In pratica si tratta di una serie di indicatori socio-economici relativi al territorio della regione calabrese a sua volta corredato da informazioni relative alle aziende che vi operano o vi hanno operato. Questo datamart sarà utilizzato nella fase di mining per effettuare la classificazione sulle imprese calabresi in base allo stato dell'attività –attività attiva, attività cessata: verranno cercati quali sono i fattori (i valori degli indicatori) che possono incidere sulla cessazione di determinate attività imprenditoriali o su una particolare riuscita imprenditoriale in un comprensorio o comune.

4.1. Creazione datamart OSCAR

Il primo datamart consiste in una tabella contenente i valori degli indicatori calcolati per ogni comune e per ogni anno disponibile. A questo scopo abbiamo eseguito una query a Campi Incrociati: ogni riga della tabella risultante rappresenta un insieme di indicatori socio-economici, per un dato comune, per ogni anno, mentre ogni cella rappresenta il valore di questi indicatori per i vari comuni e per i

vari anni. La tabella è costituita da 2906 righe e 117 colonne ed è apparsa immediatamente molto sparsa; di seguito ne mostriamo una porzione:



TabellaCI : Table

COD_ISTAT	ANNO	Abbonam	Addetti-al	Aeroporti-	Agenti-e-s	Agenzie-tr	Agenzie-m	Agenzie-pr	Agenzie-vi	Amministr	Ammontar	Analisi-co	Assicurat	Atti-e-co
101	1991		29303											
101	1993										775088			
101	1994										838243			
101	1995	6162												497
101	1996			6	8	20	7	12	12	3		25		
101	1997													
101	1998													
101001	1991		343											
101001	1993										9585			
101001	1994										8664			
101001	1995	59												7
101001	1996			0	0	0	0	0	0	0		0	0	
101001	1997													
101001	1998													
101002	1991		254											
101002	1993										9767			
101002	1994										8946			
101002	1995	36												6
101002	1996			0	0	1	0	0	0	0		0	0	
101002	1997													
101002	1998													
101003	1991		108											
101003	1993										3608			
101003	1994										3360			
101003	1995	23												3
101003	1996			0	0	0	0	0	0	0		0	0	

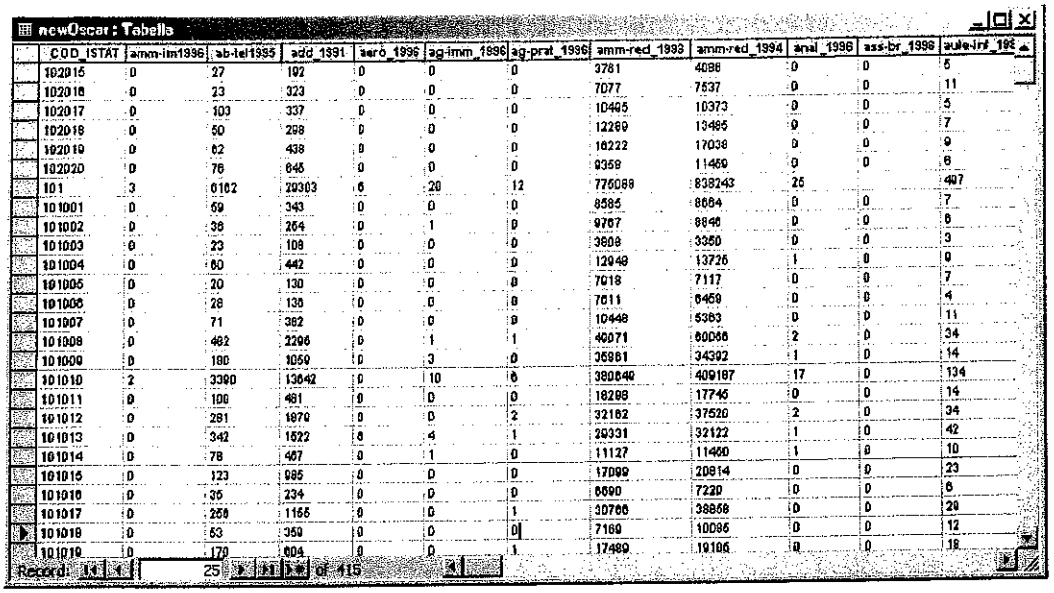
Record: 1 of 2906

Figura 4.1: Tabella CI

La tabella risultante ha rilevato le seguenti problematiche:

- Molti valori nulli, presenti in quanto ogni indicatore e' stato calcolato solo per alcuni anni;
- Estrazione della conoscenza difficoltosa: mancanza di termini di paragone tra un anno e l'altro;
- Diminuzione della performance e dell'attendibilita' dei risultati durante la procedura di mining: estrazione di regole poco significative.

Come soluzione a tali problemi abbiamo costruito una nuova tabella a Campi Incrociati in cui spostare l'informazione *Anno* dalle righe alle colonne associando l'informazione temporale agli stessi indicatori, ottenendo una tabella piu' compatta (Fig. 4.2) e utilizzabile in vista di una integrazione con i dati relativi alle imprese.



newOscar : Tabella

COD_ISTAT	amm-im1996	ab-tel1995	add_1991	aero_1996	ag-imm_1996	ag-prat_1996	amm-red_1993	amm-red_1994	anal_1996	ass-br_1996	aule-inf_199
102015	0	27	192	0	0	0	3781	4086	0	0	5
102016	0	23	323	0	0	0	7077	7537	0	0	11
102017	0	103	337	0	0	0	10495	10373	0	0	5
102018	0	50	298	0	0	0	12269	13485	0	0	7
102019	0	62	438	0	0	0	16222	17038	0	0	9
102020	0	76	646	0	0	0	9359	11469	0	0	6
101	3	6102	29303	6	20	12	776088	838243	25		497
101001	0	69	343	0	0	0	8585	8684	0	0	7
101002	0	36	264	0	1	0	9767	8846	0	0	6
101003	0	23	109	0	0	0	3608	3350	0	0	3
101004	0	60	442	0	0	0	12948	13726	1	0	9
101005	0	20	130	0	0	0	7018	7117	0	0	7
101006	0	28	138	0	0	0	7611	6459	0	0	4
101007	0	71	382	0	0	0	10448	5383	0	0	11
101008	0	462	2296	0	1	1	40071	60066	2	0	34
101009	0	180	1059	0	3	0	35981	34392	1	0	14
101010	2	3390	13642	0	10	6	380648	409187	17	0	134
101011	0	109	481	0	0	0	16208	17745	0	0	14
101012	0	281	1979	0	0	2	32182	37520	2	0	34
101013	0	342	1622	8	4	1	29331	32122	1	0	42
101014	0	78	467	0	1	0	11127	11460	1	0	10
101015	0	123	985	0	0	0	17099	20814	0	0	23
101016	0	35	234	0	0	0	6690	7220	0	0	6
101017	0	258	1166	0	0	1	30766	38858	0	0	20
101018	0	53	350	0	0	0	7169	10086	0	0	12
101019	0	170	904	0	0	1	17489	19106	0	0	18

Record: 25 of 415

Figura 4.2: Tabella finale ottenuta dalla sorgente Oscar

4.2. Creazione datamart OSCAR+ERICA

L'obiettivo di questa fase e' quello di ottenere una tabella relazionale dove ogni tupla raccolga le informazioni relative ad una impresa (parte ERICA) e al suo territorio di appartenenza (parte OSCAR). Ovviamente il lavoro svolto precedentemente sul data warehouse OSCAR non viene gettato e rappresenta quindi il nostro punto di partenza.

Il corpo centrale della sorgente Erica è costituito dalla tabella DWPTC_IMPRESA che racchiude le informazioni anagrafiche delle imprese calabresi in corrispondenza di una precisa data di monitoraggio. I record di questa tabella, quindi, descrivono eventi legati alle imprese, ad esempio per una stessa impresa che abbia cambiato il tipo di attività sono stati memorizzati due diversi record che registrano l'uno le caratteristiche di partenza dell'impresa e l'altro il cambiamento avvenuto.

Poiché non siamo interessati ad uno studio di eventi ci siamo prefisse il task di ottenere per ogni impresa un unico record che descriva lo stato più recente dell'impresa stessa. Nel seguito descriveremo i vari passi che hanno portato alla costruzione di tale tabella.

Un primo lavoro che la tabella DWPTC_IMPRESA ha richiesto e' stato l'eliminazione di alcuni campi vuoti e altri ridondanti.

Nel lavoro successivo sono stati individuati i campi che identificano univocamente una impresa. Dopo una serie di query abbiamo dedotto che fossero i seguenti:

- NREA –numero repertorio economico amministrativo.
- PROLOC –numero dell'unità locale dell'impresa, se "zero" indica la sede dell'impresa.
- CCIAA –sigla provincia della camera di commercio.
- COD_FIS –codice fiscale.
- DESC_ATT –descrizione dell'attività.

Di seguito mostriamo la formulazione di una query di selezione per ottenere, per una determinata impresa, i record rappresentanti gli eventi ad essa associati:

```
SELECT DWPTC_IMPRESA.*
FROM DWPTC_IMPRESA
WHERE (((DWPTC_IMPRESA.NREA)=348938)
        AND
       ((DWPTC_IMPRESA.PROLOC)=156)
        AND
       ((DWPTC_IMPRESA.CCIAA)="BO")
        AND
       ((DWPTC_IMPRESA.COD_FIS)="03648050015")
        AND
       ((DWPTC_IMPRESA.DESC_AT)="OPERAZIONI DI
         LOCAZIONE FINANZIARIA"));
```

Tra i record che descrivono una singola impresa, siamo interessati solo a quello relativo all'evento finale che corrisponde al record con il campo FLAG_MON diverso dal valore "R", vediamo il significato di tale campo: il campo FLAG_MON, flag di monitoraggio, può assumere i seguenti valori:

- **I** per indicare una impresa nuova iscritta.
- **C** per indicare una impresa cessata.
- **M** per indicare una impresa modificata.
- **R** per indicare una impresa regolare.

Con una semplice query di selezione sono stati formati due insiemi di record, quelli che presentano il campo FLAG_MON con il valore **R**, e quelli che presentano il campo FLAG_MON con il valore **C**, **M** o **I**; sono state cioè isolate le

imprese regolari dall'insieme delle imprese che sono cessate o hanno subito una modifica o sono nuove iscritte ottenendo due insiemi *non disgiunti*, in quanto imprese registrate come regolari e in seguito modificate cadono in entrambi gli insiemi. Il primo insieme e' stato chiamato imp_R e il secondo imp_CMI.

Il nostro scopo è quello di ottenere una partizione dei record che descrivono quelle imprese per cui è stato memorizzato un unico evento e dei record che descrivono quelle imprese per cui sono stati memorizzati più eventi. Per ottenere l'insieme dei record che descrivono imprese per cui non è stato registrato alcun cambiamento, è stato sufficiente togliere da imp_R i record riguardanti le imprese memorizzate anche in imp_CMI, applicando la seguente operazione insiemistica:

Imp_R– (imp_R ∩ imp_CMI).

L'insieme risultante unito all'insieme dato da imp_CMI rappresenta la partizione cercata e costituisce l'insieme dei record descriventi ognuno il solo evento o l'ultimo evento di una singola impresa. Il motivo per cui siamo interessati, per ogni impresa, a tale record, e' che l'ultimo evento rappresenta lo stato attuale della impresa: il nostro scopo quindi consiste nel generare una tabella che rappresenta, per ogni impresa, l'ultimo evento. La query per ottenere l'operazione *Imp_R – (imp_R ∩ imp_CMI)* ha la seguente forma:

```
SELECT *
FROM [imp_R]
WHERE NOT EXIST
      SELECT [imp_CMI].COD_FIS
      FROM imp_CMI
      WHERE ((([imp_R].CCIAA)=([imp_CMI]. CCIAA))
                AND (([imp_R].NREA)=([imp_CMI]. NREA))
          AND (([imp_R].PROLOC)=([imp_CMI]. PROLOC))
          AND (([imp_R].COD_FIS)=([imp_CMI]. COD_FIS))
          AND (([imp_R].DESC_AT)=([imp_CMI]. DESC_AT)));
```

L'unione della tabella creata dal risultato della query precedente con la tabella imp_CMI è stata ottenuta con la seguente query di accodamento il cui risultato costituisce la tabella newImpresa formata da 163180 record:

```
INSERT INTO [imp_CMI]
SELECT
FROM [RisulQuery];
```

Riportiamo di seguito la tabella risultante dalle operazioni sopra descritte, tale tabella sara' utilizzata nell'operazione di join sui comuni (CD_COM) insieme ad OSCAR.

IMP: Table

NATGIU	PROLOC	STAT_ATT	CD_COM	CAUCE	PRV	DESC_AT	CAP_SOC	CATEG_ATT	DT_INIZIO
SOCIETA' IN NOME COLL	1	CESSATA	CS108	CESSAZIONE DI	CS	MACELLERIA	0	COMM-AL-DETTAGLI	1993
IMPRESA INDIVIDUALE	0	ATTIVA	CS009		CS	ATTIVITA' COMM	0	AGRICOLTURA-CAC	1984
IMPRESA INDIVIDUALE	0	ATTIVA	CS106		CS	LAVORI EDILI S'	0		
SOCIETA' IN NOME COLL	0	ATTIVA	CS015		CS	COMM AL MINU'	0	ATTIVITA-DI-SUPPOI	1996
SOCIETA' IN NOME COLL	0	ATTIVA	CS142		CS	ATTIVITA': COM	0	ALTRE-ATTIVITA-DEI	1996
IMPRESA INDIVIDUALE	0	ATTIVA	CS023		CS	ATTIVITA': FARM	0	INDUSTRIE-ALIMEN'	
IMPRESA INDIVIDUALE	0	ATTIVA	CS060		CS	ATTIVITA': ESER	0	COMM-AL-DETTAGLI	1992
IMPRESA INDIVIDUALE	0	ATTIVA	CS045		CS	COMMERCIO IN	0		
SOCIETA' IN NOME COLL	0	CESSATA	CS025	CANCELLAZION	CS		0	COSTRUZIONI	1991
IMPRESA INDIVIDUALE	0	ATTIVA	CS132		CS	ATTIVITA': COM	0	ALBERGHI-E-RISTOI	1994
IMPRESA INDIVIDUALE	0	ATTIVA	CS140		CS	ATTIVITA': RISTO	0	ALBERGHI-E-RISTOI	1990
SOCIETA' IN NOME COLL	0	ATTIVA	CS015		CS	ATTIVITA': AUTO	0	COSTRUZIONI	1994
IMPRESA INDIVIDUALE	0	CESSATA	CS047	CESSAZIONE DI	CS	ATTIVITA': CONF	0	ALBERGHI-E-RISTOI	1996
IMPRESA INDIVIDUALE	1	ATTIVA	CS025		CS	ATTIVITA': ATTIV	0	AGRICOLTURA-CAC	1983
IMPRESA INDIVIDUALE	0	ATTIVA	CS040		CS	ATTIVITA': COM	0	AGRICOLTURA-CAC	1993
IMPRESA INDIVIDUALE	0	ATTIVA	CS070		CS	ATTIVITA': IMPR	10000000	AGRICOLTURA-CAC	1994
SOCIETA' A RESPONSAE	1	ATTIVA	CS160		CS	COMMERCIO AL	6000000		
SOCIETA' A RESPONSAE	2	ATTIVA	CS048		CS	COMMERCIO AL	6000000	TRASPORTI-TERRES	1977
SOCIETA' A RESPONSAE	0	ATTIVA	CS029		CS	COMMERCIO AL	6000000	FABBR-LAVORAZIO!	
IMPRESA INDIVIDUALE	0	ATTIVA	CS105		CS		0	INDUSTRIE-ALIMEN'	1978
IMPRESA INDIVIDUALE	0	ATTIVA	CS044		CS		0	COMM-AL-DETTAGLI	1980
IMPRESA INDIVIDUALE	0	CESSATA	CS028	CESSAZIONE DI	CS	ATTIVITA': RISTO	20000000	ALBERGHI-E-RISTOI	1978
IMPRESA INDIVIDUALE	0	CESSATA	CS046	CESSAZIONE DI	CS	ATTIVITA': LAVO	10000000	AGRICOLTURA-CAC	1979
IMPRESA INDIVIDUALE	0	CESSATA	CS086	CESSAZIONE DI	CS		0	COMM-AL-DETTAGLI	1977
IMPRESA INDIVIDUALE	0	CESSATA	CS034	MORTE	CS		0	COMM-AL-DETTAGLI	1977
IMPRESA INDIVIDUALE	0	CESSATA	CS114	CESSAZIONE DI	CS		0	COSTRUZIONI	
IMPRESA INDIVIDUALE	0	CESSATA	CS102	CESSAZIONE DI	CS		0	COSTRUZIONI	1981
IMPRESA INDIVIDUALE	0	CESSATA	CS056	CESSAZIONE DI	CS	ATTIVITA': LAVO	10000000	INDUSTRIE-ALIMEN'	1981

Record: 14 10

Figura 4.3: Esempio dei contenuti della tabella Impresa

4.3. Analisi e problema dei dati

Per conoscere l'andamento dei valori assunti da ogni attributo dei datamart, sono state calcolate alcune statistiche descrittive, estendendo, laddove necessario, le funzionalita' offerte da Weka. Tale lavoro ha consentito l'eliminazione di attributi insignificanti e l'individuazione di outliers.

Un problema riscontrato e' stato la distribuzione dei dati:

- Molti attributi assumono moda zero;
- Molti attributi del datamart Oscar seguono questo comportamento: hanno la maggior parte dei valori distribuiti uniformemente sul 90% della retta mentre nell'ultimo 10% assumono valori con un comportamento atipico, valori che sono quindi da considerarsi outliers.
- Dati di tipo numerico

Come soluzione a tali problemi abbiamo proposto la discretizzazione dei dati necessaria anche per l'estrazione di regole di associazione.

4.4. Discretizzazione

L'andamento non uniforme dei valori assunti dagli attributi rivelato dallo studio precendente ha evidenziato l'inadeguatezza del discretizzatore di Weka, equal-width binning: tale metodo, creando intervalli di uguale ampiezza, distribuisce in modo non uniforme le istanze. Alcuni bin risultano contenere molte istanze mentre altri nessuna, danneggiando, in questo modo, l'estrazione di regole di associazione significative e la costruzione di una buona struttura di mining.

E' stato quindi necessario realizzare nel linguaggio JAVA una classe che implementasse un metodo di discretizzazione più adatto alle caratteristiche dei nostri dati: il metodo *equal-frequency binning*. Tale metodo crea intervalli di differente ampiezza scelti in modo che lo stesso numero di tuple cada in ognuno di essi, [FI93].

5. ANALISI ESPLORATIVA

In questa sezione descriveremo il lavoro svolto durante l'analisi esplorativa in cui sono stati usati vari strumenti di WEKA applicati ad alcune viste di Oscar e di Oscar+ERICA. L'applicazione di tali strumenti non è stata effettuata sugli interi datamart Oscar e Oscar+ERICA per ridurre i tempi di esecuzione che risulterebbero troppo lunghi, e per indirizzare l'analisi solo su aspetti interessanti.

L'obiettivo di questa analisi consiste nel raggiungere una conoscenza più accurata su peculiarità nascoste o insite nei nostri dati che possa facilitare il lavoro svolto nella analisi predittiva.

5.1. Scelta dei task di analisi e esperimenti

Il primo task di analisi prefissato, consiste nell'individuare insiemi di comuni rappresentanti gruppi omogenei basati su metriche di similarità tra i dati, e, in un secondo passo, nell'ottenere una descrizione di ogni gruppo trovato, che fosse di facile interpretazione. Il secondo task prefissato consiste nell'estrarre relazioni tra gli attributi dei comuni, corrispondenti ai campi della tabella Oscar, con lo scopo di derivare delle correlazioni significative. Per individuare i cluster dei comuni è stato

utilizzato l'algoritmo di clustering *EM*, mentre per ottenere una descrizione di tali cluster è stato usato, il decision list *PART*. Le relazioni tra attributi dei comuni sono state ottenute applicando l'association rule learner *APRIORI*.

Vediamo in dettaglio in che modo sono stati raggiunti i due task di mining.

5.1.1. Primo task: clusterizzazione e classificazione

Abbiamo applicato ai datamart l'algoritmo *EM* ottenendo per ogni tupla l'etichetta del cluster di appartenenza. Per ottenere una descrizione di ogni cluster abbiamo applicato il classificatore *PART* eseguendo le seguenti operazioni:

- Abbiamo aggiunto al datamart un nuovo attributo, *ETICHETTE*, facendogli assumere, in corrispondenza di ogni istanza, il valore dell'etichetta del cluster assegnato dall'algoritmo *EM*;
- Abbiamo indotto le regole decisionali per il dataset risultante al passo precedente tramite applicazione dell'algoritmo *PART* assumendo come *target* l'attributo aggiunto.

Tali passi sono illustrati nella figura seguente.

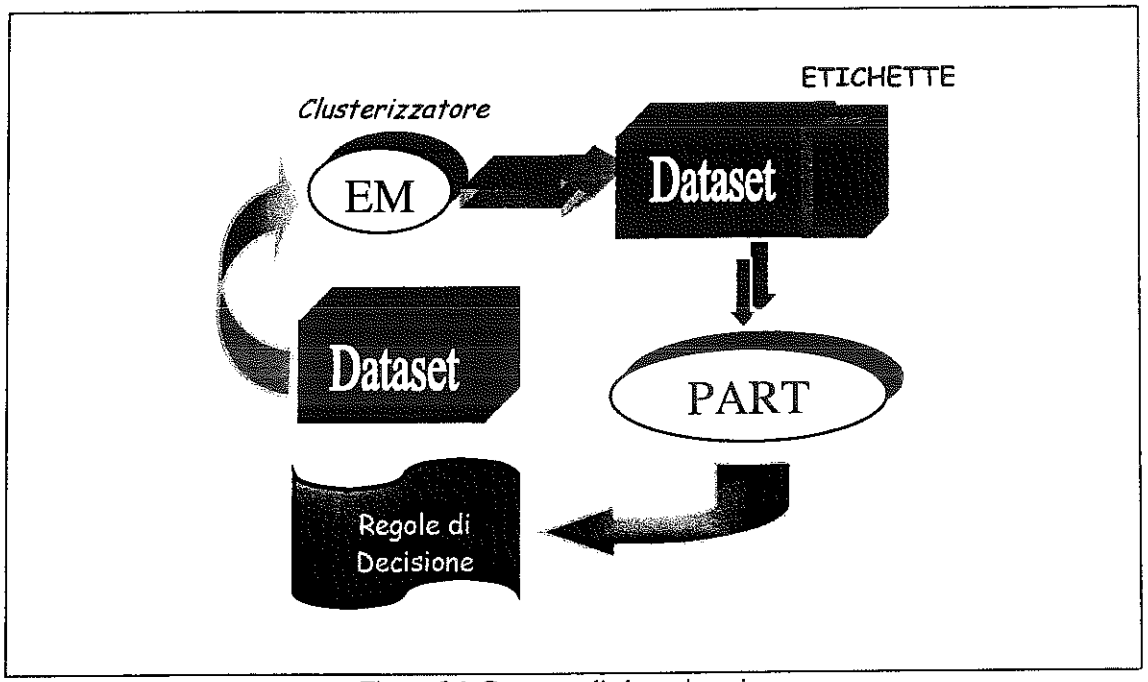


Figura 5.1: Processo di clusterizzazione

Le informazioni sulle peculiarità di ogni cluster sono ottenute analizzando l'insieme di regole di decisione che hanno lo stesso valore della variabile target e che quindi descrivono uno stesso cluster.

Automazione del processo. Tali strumenti sono stati estesi e integrati, a seconda dell'analisi in corso, al fine di realizzare un learner che automatizzi il processo sopra descritto.

Esperimento 1: clustering e classificazione

Un primo esperimento svolto consiste nell'analizzare il datamart ottenuto selezionando dal dataset Oscar alcuni attributi in base alle seguenti valutazioni:

- Scelta di attributi significativi perche' coprenti vari aspetti dei comuni: aspetti demografici, produttivi ed economici;
- Scelta di attributi aventi un numero basso di valori nulli;
- Scelta di attributi aventi una distribuzione dei valori abbastanza uniforme.

I comuni sono stati raggruppati per soglie differenti di: Dimensione, Vocazione Turistica, Vivacità e Benessere Economico, Connessione alle Reti di Trasporto.

Tra tutti i cluster ottenuti mostriamo un esempio.

Regola descrivente il cluster A:

```
Vocazione turistica nel 1994 <= 3.3
AND
Concentrazione delle attivita' produttive nel 1991 <= 0.161
```

Regola descrivente il cluster B:

```
Vocazione turistica nel 1994 > 6.4
AND
Concentrazione delle attivita' produttive nel 1991 > 0.161
AND
Connessione alle reti di trasporto nel 1991 > 44
```

Il primo cluster raggruppa comuni economicamente meno vivaci rispetto a quelli identificati nel secondo cluster.

Esperimento 2: clustering e classificazione

Il secondo esperimento svolto consiste nell'analizzare il datamart ottenuto selezionando dal dataset Oscar alcuni attributi economici per analizzare gli aspetti produttivi del territorio in vista dello studio predittivo dell'andamento delle attività di una impresa in base al comune in cui risiede. Come risultato abbiamo ottenuto quattro cluster in cui i comuni sono stati raggruppati in base alla tipologia di ricchezza e di consumo. Tra tutti i cluster ottenuti mostriamo un esempio.

Regola descrivente il cluster A:

```
Numero dei contribuenti IRPEF nel 1994 > 2283
AND
```

Depositi aziende di credito_1996 > 784867

Regola descrivente il cluster B:

Utenze telefoniche private nel 1997 = [1460,3249]
AND
Autovetture circolanti con cilindrata oltre i 2000 cc nel 1994
<= 80

E' interessante notare che concentrando l'attenzione su aspetti economici abbiamo avuto risultati differenti rispetto all'esperimento precedente: i comuni, in questo caso, sono stati raggruppati per soglie di ricchezza, ad esempio per un alto numero di contribuenti IRPEF e per alti depositi delle aziende di credito.

5.1.2. Secondo task: estrazione di regole associative

Abbiamo applicato ai datamart l'algoritmo *Apriori*. Poiché tale algoritmo lavora su dati discreti, prima di applicarlo abbiamo discretizzato i datamart utilizzando l'algoritmo di discretizzazione da noi implementato e descritto nella sezione precedente.

Automazione del processo. Tali strumenti sono stati estesi e integrati, a seconda dell'analisi in corso, al fine di realizzare un learner che automatizzi il processo sopra descritto.

Esperimento 1: discretizzazione e estrazione di regole associative

Un primo esperimento svolto in questa fase, consiste nell'analizzare il datamart ottenuto selezionando dal dataset Oscar alcuni attributi in base alle seguenti valutazioni:

- Scelta di attributi significativi perche' coprenti vari aspetti dei comuni: aspetti demografici, produttivi ed economici;
- Scelta di attributi aventi un numero basso di valori nulli;
- Scelta di attributi aventi una distribuzione dei valori abbastanza uniforme.

Come risultato sono state estratte 50 regole, di cui 4 significative.

Diamo di seguito un esempio di regola estratta:

Indicatore-benessere-econ_1995='[56.0-69.1]'
Perc-total-reg-di-op-ec-telem-sensibili_1996='[0.01-0.02]'
⇨ *Vivacita-economica_1996='[0.0040-0.0080]'*

che caratterizza comuni molto poveri e poco vivaci economicamente.

Esperimento 2: discretizzazione e estrazione di regole associative

Se con il primo esperimento abbiamo cercato di coprire vari aspetti del territorio, in questo secondo esperimento abbiamo concentrato l'analisi su aspetti produttivi in vista di uno studio predittivo sulle imprese. Sono state estratte 100 regole di cui 14 significative. Mostriamo di seguito un esempio di regola estratta:

Consumi-en-el-non-res-1997='[1298.0-1798.0]'
Indicatore-sintesi-tenore-di-vita-1995=[114.0-128.0]
⇨ *N-contrib-red-impon-sup-40-ml-1994=[1942.0-4778.0]*

che caratterizza comuni abbastanza ricchi.

5.2. Valutazione e risultati

In seguito a tali esperimenti concludiamo che abbiamo ottenuto dei buoni risultati per quanto riguarda il processo di clusterizzazione: i raggruppamenti si sono rivelati significativi e di facile interpretazione.

Inoltre i risultati ottenuti possono condurre ad applicazioni molto interessanti:

- Puo' essere fatto un monitoraggio (temporale) delle popolazioni dei cluster, ad esempio se uno stesso esperimento viene condotto negli anni successivi, confrontando i risultati e' possibile individuare eventuali variazioni avvenute.
- Puo' essere condotto uno studio di outliers: se l'analisi di un certo cluster porta ad evidenziare la presenza di un comune le cui caratteristiche si discostano da quelle degli altri comuni appartenenti allo stesso cluster, tale outlier puo' essere argomento di studio per rivelare o un'anomalia dei dati o interessanti peculiarita' sui dati stessi.

Diversamente le regole di associazione sono risultate poco soddisfacenti, in quanto hanno evidenziato caratteristiche piuttosto ovvie e poco significative. Una possibile soluzione a questo problema consiste nel combinare l'estrazione di RdA con altre tecniche, ad esempio con la clusterizzazione .

6. ANALISI PREDITTIVA

In questa sezione sono descritti i risultati di analisi di tipo predittivo svolta sull'unione dei dati economici relativi alle imprese calabresi (ERICA) con gli indicatori socio-economici del territorio di appartenenza, con l'obiettivo di:

- Spiegare le cause di cessazione di una certa impresa in termini delle caratteristiche del territorio di appartenenza;
- Creare un servizio informativo per l'individuazione delle zone del territorio calabrese più promettenti per una nuova impresa.

L' idea di base è quella di costruire degli alberi di classificazione allenati a distinguere i profili delle imprese ancora attive (che dovrebbero rappresentare imprese in buona salute) da quelle che hanno cessato l'attività. Tali profili saranno formati non solo da dati economici riguardanti le imprese stesse, ma anche dagli indicatori socio-economici del territorio di appartenenza.

6.1. Primo esperimento: applicazione C4.5 al datamart Oscar+ERICA

Il primo esperimento consiste nell'applicazione dell'algoritmo C4.5, di WEKA al datamart contenente informazioni relative ad una impresa e al suo territorio di appartenenza. L'obiettivo è di derivare un predittore in grado di distinguere i profili delle imprese ancora attive da quelle che hanno cessato l'attività. La variabile target della nostra analisi è quindi lo STATO ATTIVITA' formato da due classi: ATTIVA e CESSATA. L'albero di fig. 6.1 descrive il risultato di questo esperimento.

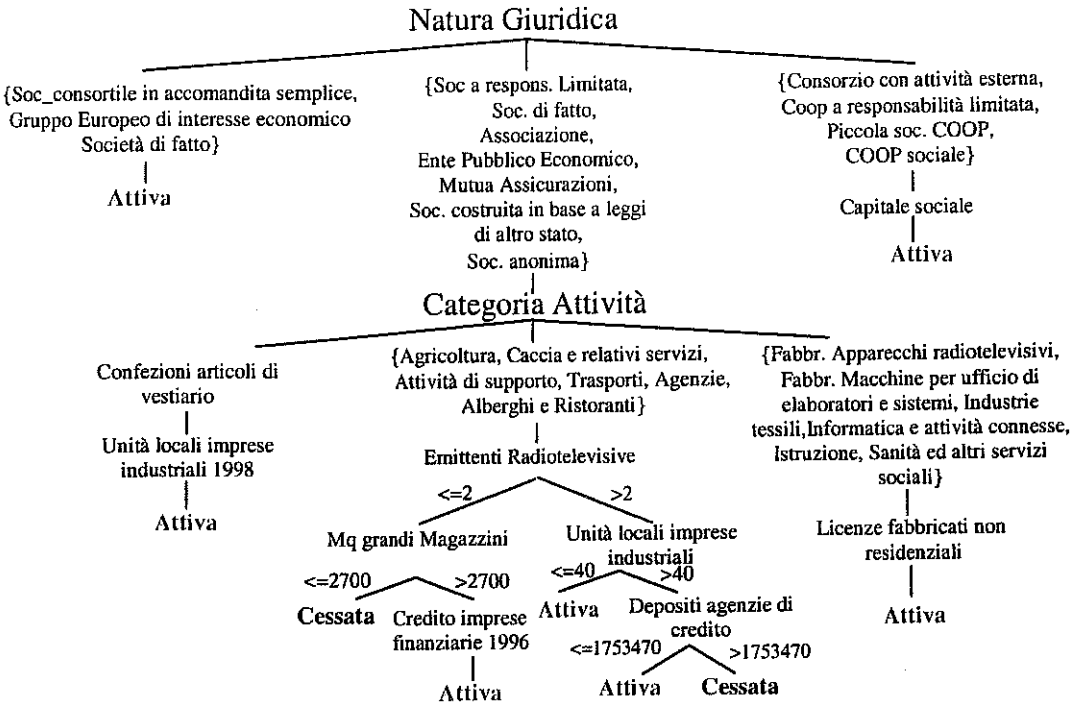


Figura 6.1: Esempio di albero ottenuto applicando il C4.5 all'intero dataset

Il modello estratto mette in evidenza le seguenti peculiarita':

- I primi attributi di split sono Natura Giuridica e Categoria Attività: quindi proprio i due attributi direttamente collegati alle imprese risultano essere i più informativi al fine della classificazione Attiva/Cessata. E' ragionevole pensare che una sorgente informativa piu' ricca di informazioni relative alle imprese (ad esempio contenente gli indici di bilancio, il numero addetti, etc) permetterebbe la costruzione di profili molto affidabili.
- L'alta percentuale delle foglie dell'albero è etichettata come *Attiva*, questo perché l'algoritmo si specializza al dataset di allenamento formato per il 93% da istanze aventi l'attributo target uguale ad attiva.

Il modello dell'albero sopra descritto è stato ottenuto riassumendo la visualizzazione testuale, messa a disposizione da WEKA, per motivi di spazio sono stati riportati solo i valori di alcuni attributi.

6.2. Interpretazione di un profilo

Seguendo il cammino dalla radice ad una foglia, troviamo un profilo di imprese prevalentemente cessate:

if NATURA-GIU = PICCOLA-SOC-COOP-A-RESPONS-LIMITATA
and CATEG-ATT = ALBERGHI-E-RISTORANTI
and Emittenti Radiotelevisive<=2
and Mq-grandi–magazzini<=2700
then Classe = CESSATA

Tale profilo corrisponde allo 0,1% dei casi totali, ma il 93% delle imprese che cadono in tale profilo sono effettivamente cessate. E' interessante notare la correlazione tra le informazioni relative all'impresa (NATURA-GIU, CATEG-ATT) e quelle relative al territorio (Emittenti Radiotelevisive, Mq-grandi-magazzini) nella classificazione. Volendo interpretare questo profilo possiamo supporre che in comuni particolarmente piccoli e con scarsa vocazione turistica le imprese con categoria di attività alberghi e ristoranti tendano a cessare. Infatti è ragionevole pensare che la superficie dei grandi magazzini sia abbastanza correlata con la popolazione del territorio e con la vocazione turistica. Dove ci sono molti abitanti oppure turismo ci sono anche grandi magazzini.

6.3. Valutazione modello

L'albero generato dall'applicazione dell'algoritmo C4.5, [Q93], al dataset avente la distribuzione originaria della variabile target risulta essere molto accurato in quanto individua informazioni di nicchia particolareggiate. Tuttavia tale modello risulta poco utile e specifico per ottenere predizioni attendibili sulla variabile target (Attiva/Cessata), dato che l'algoritmo si specializza sul dataset di allenamento considerando le istanze aventi l'attributo target uguale a Cessata come rumore, come rileva la seguente matrice di confusione avente sulla riga relativa alla classe effettiva Attiva un notevole sbilanciamento.

		Classe predetta	
		attiva	*cessata*
Classe effettiva	*attiva*	8249	9
	cessata	516	25

Tabella 6.1: Matrice di confusione relativa C4.5

La sezione successiva tratta dettagliatamente questo problema.

6.4. Creazione dataset bilanciati

Molti fattori contribuiscono al successo di un processo di learning,: un fattore determinante è la distribuzione di classe nell'insieme di training. Sperimentalmente si puo' notare che utilizzando lo stesso algoritmo di learning, su insiemi di training contenenti una differente distribuzione di classe si possono generare classificatori di differente qualità; utilizzare, perciò, la "*naturale*" distribuzione di classe potrebbe non generare il classificatore più efficace. In particolar modo, alcuni algoritmi di learning, in presenza di dati altamente sbilanciati trattano la "classe di minoranza" come "*rumore*", generando un classificatore che predice sempre la "classe di maggioranza".

Nell'esperimento precedente abbiamo riscontrato questo problema nel dataset Oscar+ERICA, che e' costituito dal 7.7% di istanze aventi l'attributo target uguale a Cessate. Tale dataset, il cui attributo target assume valore {attiva, cessata}, in letteratura chiamato dataset two-class, presenta quindi un elevato sbilanciamento. Applicando semplicemente l'algoritmo C4.5 abbiamo osservato che il modello risultante, in questo caso l'albero decisionale, classifica tutte le istanze test come attive e ne deriva quindi un predittore poco utile seppur accurato.

Per aumentare l'abilita' dello schema di learning applicato, abbiamo combinato un meta classificatore che unisce l'output di altri learner, detti di base, con la costruzione di un insieme di particolari training set su cui addestrare il meta learner. L'approccio consiste nel creare dei sottoinsiemi con la desiderata

distribuzione di classe (50:50), *dataset bilanciati,* senza rimuovere alcun dato dall'insieme dei dati di partenza [PC, SS98].

Per esempio, se la naturale distribuzione dei dati è 20:80 e la desiderata distribuzione è 50:50, si dividono le istanze il cui attributo di classe è maggioritario in 4 partizioni, si formano 4 sottoinsiemi unendo le istanze contenenti la classe minoritaria con ognuna delle 4 partizioni contenenti la classe maggioritaria. Il processo è illustrato in figura 6.2.

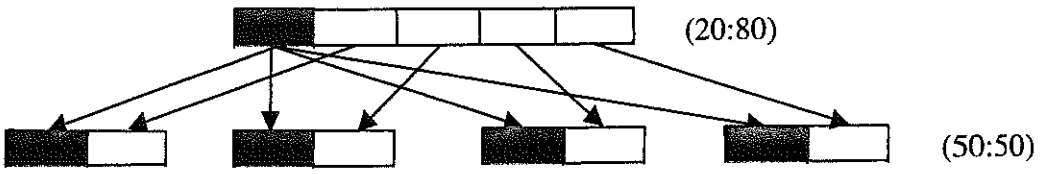


Figura 6.2: Generatore di 4 sottoinsiemi aventi una distribuzione di classe pari a 50:50, ottenuto da Un dataset avente la distribuzione di classe pari a 20:80.

Per tale realizzazione non e' stato possibile utilizzare i meta classificatori messi a disposizione da Weka, *Bagging, AdaBoostM1* e *Stacking*, per i seguenti motivi:

1. Il Bagging crea delle partizioni dei dati casuali; non essendoci nessun metodo che induca tali algoritmi a porre attenzione sulla distribuzione dell'attributo di classe, l'approccio offerto dalla libreria è risultato inadeguato, e da qui e' sorta la necessità di sviluppare un sistema di classificatori *ad hoc*.
2. Il AdaBoostM1 utilizza un meccanismo di pesatura che è fissato, mentre in questo caso vogliamo definire un meccanismo *ad hoc*.
3. Stacking: non è rilevante utilizzare algoritmi differenti, e comunque il problema del partizionamento rimane.

Il nostro task consisteva nel partizionare il dataset ATTIVE in 50 parti (ottenute dividendo il numero di istanze attive per il numero di istanze cessate) e unire ognuno di questi insiemi con il dataset CESSATE, ottenendo in questo modo 50 dataset bilanciati.

6.5. Scelta degli attributi

La scelta degli attributi su cui fare training è stata compiuta dopo una fase di pulizia e data-preparation:

- Si sono scartati gli attributi aventi molti valori mancanti;
- Abbiamo scelto 15 campioni tra i 50 dataset bilanciati (Fig 6.2), su ognuno di questi e' stato applicato l'algoritmo C4.5 selezionando solo quegli attributi presenti nei modelli estratti, essendo i piu' informativi al fine della classificazione Attiva/Cessata.

6.6. Costruzione meta learner con voto

Data l'impossibilità di utilizzare l'algoritmo C4.5, il secondo esperimento consiste nello sviluppare un sistema di classificatori *ad hoc*, capaci di predire la cessazione o meno delle attività di una impresa presente sul territorio calabrese. Definiti i dati di input bilanciati abbiamo creato la struttura del dataset inizialmente vuoto (m_data), formato da 8799 righe e da 50 colonne (una per ogni modello), in aggiunta, due altre colonne: una contenente l'identificatore di tupla e una contenente la classe effettiva. Riportiamo di seguito lo schema dell'algoritmo Newbagging.

Crea un dataset *m_data*, inizializzato a *null*
Per ognuno degli *n* dataset di training bilanciati
applica l'algoritmo C4.5
memorizza il modello risultante.
Per ogni tupla *t* contenuta nel dataset di testing
ogni modello *n* predice la classe dell'istanza
memorizza le predizioni nel dataset *m_data*

Tabella 6.2: Algoritmo Newbagging

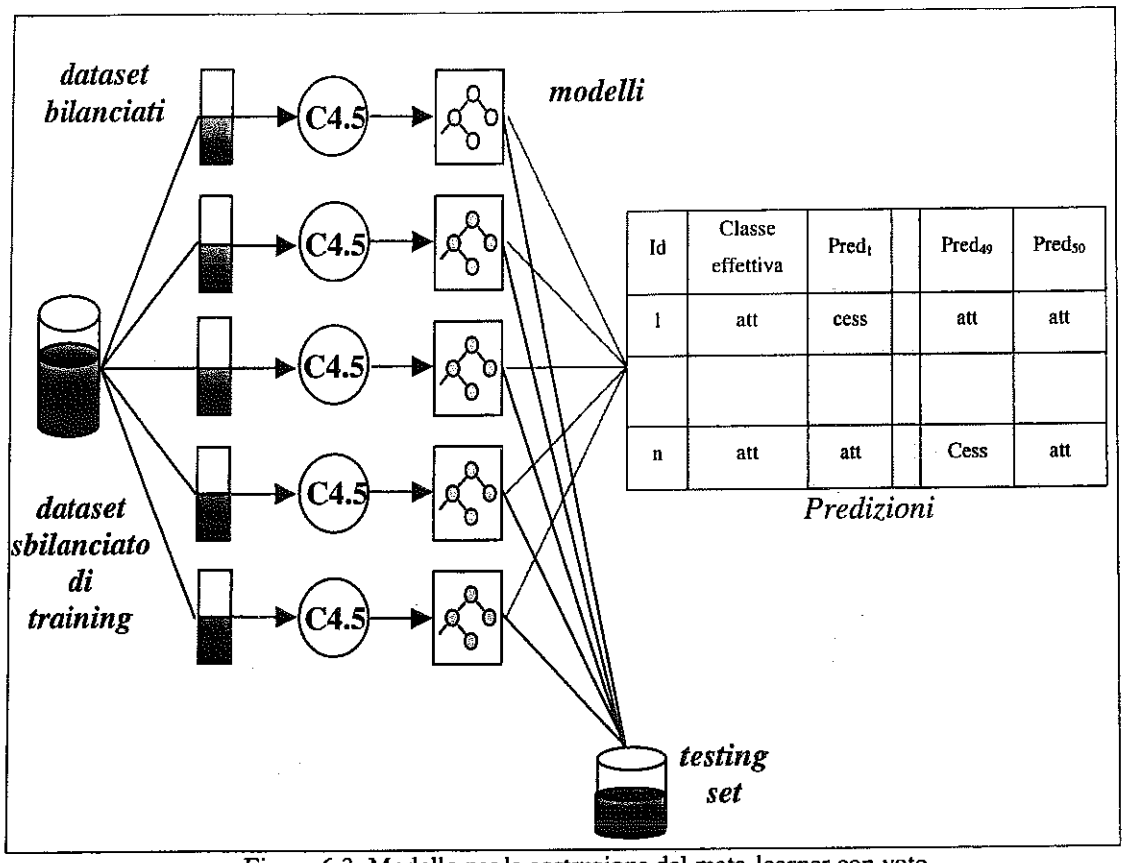


Figura 6.3: Modello per la costruzione del meta-learner con voto

La Fig. 6.3 riassume i passi dell'algoritmo Newbagging implementato.

6.7. Costruzione della matrice di confusione relativa al metalearner con votazione

Il metalearner con votazione restituisce come risultato un dataset con un numero di istanze pari al testing set costituito da 8799 tuple. Ogni istanza è costituita da un identificatore numerico, dalla classe effettiva, e dalle 50 predizioni dei modelli costruiti dal metalearner. La classe predetta dal meta-learner con voto è calcolata nel modo seguente: ogni modello predice la classe dell'istanza test e l'istanza viene classificata con la classe che ha ricevuto maggiori voti (vince la maggioranza). In Fig. 6.4 è illustrato tale processo.

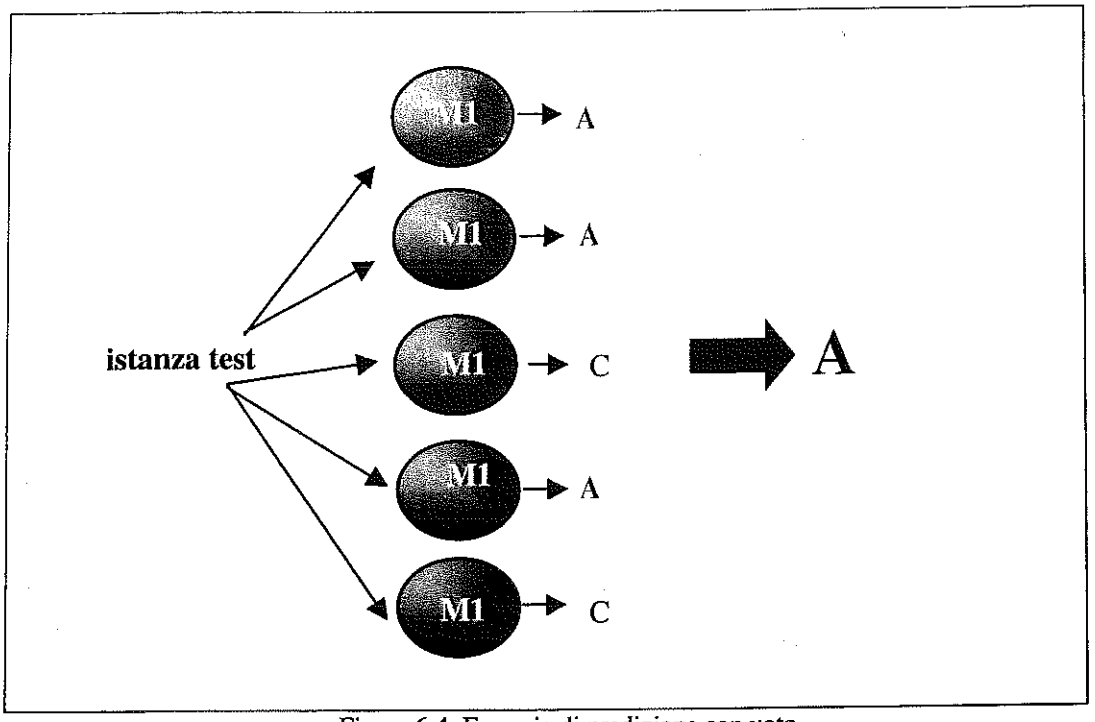


Figura 6.4: Esempio di predizione con voto

La matrice di confusione risultata da tale processo e' rappresentata in tabella 6.3, per una valutazione si rimanda alla sezione successiva.

		Classe predetta	
		attiva	*cessata*
Classe effettiva	*attiva*	5690	2568
	cessata	395	146

Tabella 6.3: Matrice di confusione ottenuta con il meta-learner con voto

6.8. Costruzione meta learner con votazione pesata

Il meta learner presentato nella sezione precedente e' caratterizzato dal fatto di attribuire ad ogni modello estratto lo stesso peso. In questo paragrafo, proponiamo la costruzione di un meta learner differente, implementato sul modello del *Winnow Algorithm* [SS]. Tale algoritmo prende in input le predizioni del meta learner con voto associando ad ognuna di esse un peso. Inizialmente il vettore dei pesi la cui dimensione e' pari al numero delle predizioni, è settato ad uno. Ogni qualvolta si

incontra una misclassificazione, i pesi associati alle predizioni vengono modificati nel seguente modo:

- Se l'algoritmo predice Attiva su un esempio Cessata, per ogni predittore che predice Cessata, si raddoppia il valore del peso.
- Se l'algoritmo predice Cessata su un esempio Attiva per ogni predittore che predice Cessata, si dimezza il valore del peso.

Lo schema del codice dell'algoritmo è illustrato nella tabella 6.4:

Winnow Algorithm
Inizializza il vettore dei pesi $w_1,\ldots..,\ w_n$ ad 1 Dato il vettore delle predizioni $x = \{x_1,\ldots..,\ x_n\}$, restituisci 1 se $w_1x_1 + w_2x_2 + \ldots.. + w_nx_n \geq n$ altrimenti restituisci 0. L'algoritmo commette un errore: a. Se predice negativo su un esempio positivo, allora per ogni predittore che predice Cessata, si raddoppia il valore del peso. b. Se l'algoritmo predice positivo su un esempio negativo, allora per ogni predittore che predice Cessata, si dimezza il valore del peso. Ritorna al passo 2

Tabella 6.4: Algoritmo Winnow

È interessante osservare che l'algoritmo presta attenzione ai predittori il cui output è cessata, allo scopo di dar peso, tra tutti i classificatori, a quelli che si sono specializzati maggiormente sull'etichetta di classe minoritaria cessata.

6.9. Costruzione della matrice di confusione relativa al meta learner con pesi

Il meta learner descritto nella sezione precedente restituisce come risultato il vettore dei pesi associati ad ogni modello, oltre al numero che identifica l'istanza e la classe effettiva dell'istanza. Anche per tale meta-classificatore abbiamo calcolato la matrice di confusione per poter confrontare le prestazioni e l'affidabilità del meta-learner pesato con quello basato su votazione semplice.

La classe maggioritaria (classe predetta dal meta learner) in presenza di votazioni pesate è calcolata nel modo seguente: vengono sommati i pesi associati ai classificatori che hanno assegnato all'istanza test la stessa etichetta di classe e viene assegnata all'istanza la classe che ha ottenuto maggior peso. Utilizzando i

pesi, la classe maggioritaria sarà quella a cui verrà assegnato il peso maggiore con il procedimento precedentemente descritto.

Per ottenere la classe maggioritaria delle predizioni del meta-learner abbiamo implementato un algoritmo che calcola la classe maggioritaria y eseguendo l'operazione matriciale $y = Xw$ dove X è la matrice rappresentante l'output del meta learner con voto e w è il vettore trasposto dei pesi ottenuto. La matrice di confusione risultata da tale processo e' rappresentata in tabella 6.5, per una valutazione si rimanda al paragrafo successivo.

		Classe predetta	
		attiva	***cessata***
Classe effettiva	***attiva***	263	7995
	cessata	15	526

Tabella 6.5: Matrice di confusione ottenuta con il meta-learner winnow

6.10. Valutazione

Una volta calcolate le matrici di confusione relative ai due meta-learner siamo in grado di dare una valutazione della loro accuratezza, espressa in termini di matrice di confusione. Da una prima analisi delle matrici di confusione si nota che l'errore $\left(\frac{FN+TN}{pos+neg}\right)$, commesso dai due meta-learner è maggiore dell'errore ottenuto applicando l'algoritmo C4.5 direttamente al dataset sbilanciato. Tale risultato non è inaspettato, poiché l'albero decisionale si specializza al dataset, trattando le istanze aventi l'attributo di classe uguale a cessata come "*rumore*". I due meta-learner sono stati progettati non tanto per abbassare l'errore, quanto piuttosto per "*sensibilizzare*" i modelli sulle istanze test aventi l'attributo di classe uguale a cessata, per tale motivo non possiamo utilizzare come parametro di confronto l'errore. Le seguenti misure permettono di esprimere una stima piu' appropriata:

$$sensitivity = \frac{TP}{pos}$$

$$specificity = \frac{TN}{neg}$$

L'accuratezza dei due meta-learner è espressa in termini delle due quantità

$$accuracy = sensitivity\frac{pos}{(pos+neg)} + specificity\frac{neg}{(pos+neg)}$$

dove:

- *pos* rappresenta il numero di istanze aventi il valore dell'attributo di classe uguale a *cessata*
- *neg* rappresenta il numero di istanze aventi il valore dell'attributo di classe uguale a *attiva.*

Riportiamo di seguito i valori dei parametri sensitivity e specificity calcolati utilizzando l'archivio OSCAR+ERICA

sensitivity (meta-learner con voto) = 0,266
sensitivity (meta-learner winnow) = 0,972
sensitivity C4.5 = 0.046

Questo risultato rileva che il meta-learner winnow rispetto al meta-learner con votazione, si specializza sulle istanze con attributo di classe cessata; tale ipotesi è validata dai risultati riportati nelle matrici di confusione, difatti nella matrice A il numero di istanze cessate miscassificate, vale a dire le istanze con attributo di classe pari a cessate predette come attive, è pari a 15 contro le 395 citate nella matrice B.

Per quanto riguarda la misura specificità, i valori per i rispettivi meta-learner sono riportati di seguito:

specificity (meta-learner con voto) = 0.689
specificity (meta-learner winnow) = 0.031
specificity C4.5 = 0.998

Tali valori indicano che il meta-learner con voto presta più attenzione alle istanze aventi valore dell'attributo di classe uguale a attive, (difatti misclassifica 2568 contro le 7995 del meta-learner winnow). Tutto cio' ha suggerito che il meta learner con voto esegue una classificazione complessivamente piu' attendibile. Ultimiamo la valutazione mettendo a confronto l'*accuracy* dei due meta-learner:

accuracy (meta-learner con voto) = 0.729

accuracy (meta-learner winnow) = 0.097

I risultati avvalorano la nostra ipotesi: il meta-learner più affidabile per predire la continuità o meno delle attività di una impresa presente nel territorio calabrese, risulta essere il meta-learner con voto come la Fig 6.5 mostra.

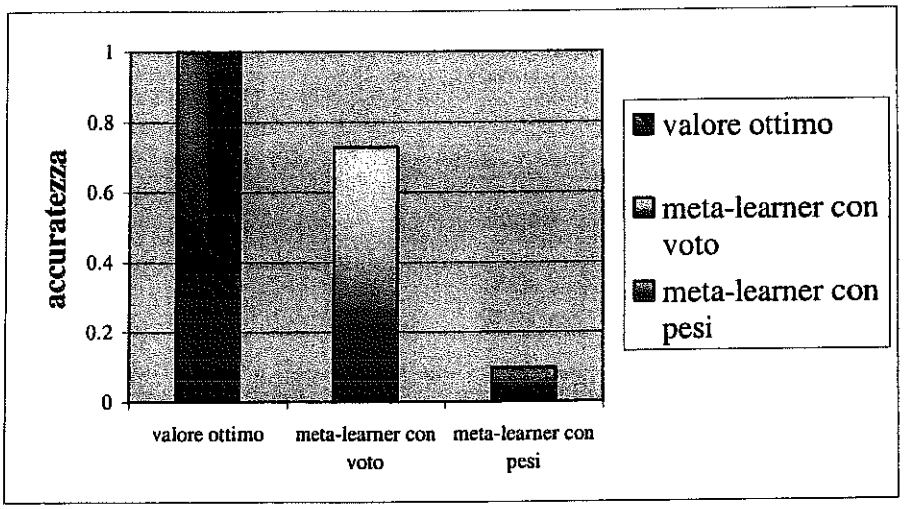


Figura 6.5: Valutazione risultati

7. CONCLUSIONI

Nella presente articolo è stato affrontato lo studio e lo sviluppo di strumenti per l'analisi dei dati contenuti nel Data Warehouse Piano Telematico Calabria (DW-PTC). Durante l'analisi di pre-processing sono stati sviluppati strumenti di calcolo di statistiche descrittive sui dati e uno strumento di discretizzazione particolarmente adatto per le caratteristiche dei dati che avevamo a disposizione. Durante l'analisi esplorativa sono stati progettati due learner per integrare diversi task, la clusterizzazione con la classificazione, e la discretizzazione con l'estrazione di regole associative, al fine di automatizzare l'intero processo esplorativo. Infine abbiamo posto l'attenzione sulla progettazione e sullo sviluppo di un sistema di classificatori in grado di predire la cessazione o la continuità di un'impresa presente sul territorio calabrese, in base alla tipologia del comune in cui l'azienda è dislocata.

Il lavoro è stato sviluppato secondo due punti fondamentali che andremo ad elencare:

a. Sviluppo di un **case study** che illustra un insieme di esperimenti effettuati sui dati del Data Warehouse Piano Telematico Calabria. Il principale obiettivo che ci siamo posti in questa fase è di applicare le vari passi del processo KDD alle informazioni suddette utilizzando tecniche di supporto al processo KDD presenti nella libreria Weka, opportunamente integrate con tecniche più consone a coprire analisi sofisticate.
b. **Analisi** e valutazione, attraverso il caso di studio, delle funzionalità di un ambiente di sviluppo per applicazioni di Data Mining basato su Java, con particolare rilievo alle capacita' di modellare il processo per raggiungere una conoscenza piu' accurata su domini applicativi particolari.

In particolare, sono state implementate tecniche di pre-processing, per il trattamento del problema di dati con una distribuzione non uniforme dei valori, e tecniche predittive per il trattamento dei dati altamente sbilanciati, ossia che hanno una differente distribuzione di classe. Quest'ultimo problema e' stato riscontrato nel dataset risultante dall'integrazione dei dati delle imprese con i dati socio-economici del territorio. A questo proposito abbiamo sviluppato un sistema di meta learning, che ribilancia le distribuzioni, e abbiamo confrontato le prestazioni del sistema con la semplice applicazione dell'algoritmo C4.5. Infine, abbiamo proposto l'implementazione di un'interfaccia per una applicazione di analisi socio-demografica dei dati DW-PTC.

Il presente lavoro è stato svolto utilizzando WEKA, Waikato Environment for Knowledge Analysis, un ambiente sviluppato dall'Università di Waikato interamente scritto in Java e reso disponibile sul Web. Tale sistema mette a disposizione tutti gli strumenti di supporto al processo KDD in modo integrato ed uniforme. La libreria WEKA pur essendo uno strumento molto versatile, presenta alcuni inconvenienti da non sottovalutare:

1. Non si interfaccia direttamente con i Data Base. L'utente deve effettuare una fase di pre-elaborazione per poter processare il file con WEKA;
2. E' main-memory, ossia prima di effettuare una qualsiasi elaborazione carica in memoria principale tutto il file, sollevando problematiche legate alla dimensione dello spazio degli attributi.

3. Il sistema WEKA è una libreria interamente scritta in Java quindi non è possibile ottenere performance eccezionali.

8. RIFERIMENTI

[CHY96] M. S. Chen, J. Han, P. S. Yu. "Data Mining: An Overview from a Database Perspective." In *IEEE Transactions on Knowledge and Data*

[FI93] U. M. Fayyad, K. B. Irani. "Multi-interval discretization of continuous-valued attributes for classification learning." *Proceedings of IJCAI, Chambery, France*, 1993.

[FPS96] U. M. Fayyad, G. Piatetsky-Shapiro e P. Smyth. "From Data Mining to Knowledge Discovery: an Overview". In *Advances in Knowledge discovery and Data Mining*, U. M. Fayyad, G.Piatetsky-Shapiro P. Smyth e R. Uthurusamy (Eds.), pages 1-34, AAAI Press, 1996.

[FW00] Eibe Frank, Ian H. Witten "Data Mining: Practical Machine Learning Tools and Tecniques with Java Implementation", 2000. Series Editor Microsoft Research.

[GH94] Geoffrey Holmes, "Weka: A Machine Learning Workbench" 1994. Department of Computer Science University of Waikato, Hamilton, New Zealand

[M96] H. Mannila. "Data Mining: machine learning, statistics and databases". In *8th Int'l Conf. on Scientific and Statistical Database Management*, Stoccolma, Giugno, pages 1-8, 1996.

[PT99] C. Pizzuti e D. Talia. "Knowledge Discovery e Data Mining: Concetti, Algoritmi e Sistemi". In *Rivista di Informatica*, vol. XXIX, no. 1, 1999. http://isi-cnr.deis.unical.it:1080/~talia/datamin.ps.

[Q93] J. R. Quinland. "C4.5: Programs for Machine Learning". San Matteo, CA: Morgan Kaufmann, 1993.

[SS] Salvatore J. Stolfo "Learning with Non-uniform Class and Cost Distribution: Effects and a Multi-classifier Approc"