Corso di laurea in Ingegneria Gestionale

Alberto Munfuletto

Progettazione e sviluppo di una metodologia di analisi di dati nel contesto di aziende metalmeccaniche

Tesi di laurea magistrale

Collaborazione con: **Lan Tek Service SRL**

Relatore: **Tania Cerquitelli**

Anno Accademico: **2018-2019**

Indice

Introduzione **6**

1. Data Selection **9**

2. Analisi visiva del dataset **13**

2.1 Analisi visiva Ordini di Vendita 13
2.2 Analisi visiva Ordini di Conto Lavoro 17
2.3 Analisi visiva Ordini di Acquisto 18

3. Preprocessing **19**

3.1 Preprocessing Ordini di Vendita 21
3.2 Preprocessing Ordini di Acquisto 22
3.3 Preprocessing Ordini di Conto Lavoro 22
3.4 Preprocessing righe degli ordini di Conto Lavoro 23
3.5 Preprocessing righe degli ordini di Vendita 24
3.6 Preprocessing righe degli ordini di Acquisto 24

4. Clustering **25**

4.1 Clustering dati strutturati 25
4.1.1 Clustering dati Ordini di Acquisto 27
4.1.2 Clustering dati Ordini di Conto Lavoro 28
4.1.3 Clustering dati Ordini di Vendita 28

4.2 Clustering di testo 29
4.1.1 Clustering testuale Ordini di Vendita 31
4.1.2 Clustering testuale Ordini di Conto Lavoro 33
4.1.3 Clustering testuale Ordini di Acquisto 35

5. Association Rules **37**

5.1 Association Rules Ordini di Vendita 38
5.2 Association Rules Ordini di Acquisto 40
5.3 Association Rules Ordini di Conto Lavoro 44

6. Classification **48**

6.1 Classification Ordini di Vendita 51
6.1.1 Cross Validation Ordini di Vendita *53*
6.1.2 Apply Model Ordini di Vendita *55*
6.1.3 Correlation Matrix Ordini di Vendita *56*

6.2 Classification Ordini di Acquisto 57
6.2.1 *Cross Validation Ordini di Acquisto* *59*
6.2.2 *Apply Model Ordini di Acquisto* *61*
6.2.3 *Correlation Matrix Ordini di Acquisto* *62*

6.3 Classification Ordini di Conto Lavoro 64
6.3.1 *Cross Validation Ordini di Conto Lavoro* *66*
6.3.2 *Apply Model Ordini di Conto Lavoro* *68*
6.3.3 *Correlation Matrix Ordini di Conto Lavoro* *69*

Conclusioni **70**

“Suerte, trabajo y silencio”
Hector Cuper

Introduzione

Ho svolto il tirocinio curricolare presso Lan Tek Service Srl. L'azienda si occupa della fornitura di software CAM e ERP e di tutte le attività post vendita riguardanti i prodotti venduti quindi l'assistenza, la consulenza e la formazione al cliente sui prodotti venduti.

I due software principali sono:

- software CAM: *Lantek Expert*
- software ERP: *Concept*

Entrambi i software sono stati sviluppati esternamente. In particolare, *Lantek Expert* è il prodotto di punta di Lantek, multinazionale con sede centrale a Bilbao. L'azienda per cui ho svolto il tirocinio è appunto Lantek Italia, la filiale nazionale dell'azienda. *Concept*, invece, è stato sviluppato da una software house piemontese, Azerouno Srl di cui Lantek è un rivenditore specializzato. Dato il mio percorso di studi durante il tirocinio sono stato inquadrato nel reparto dedicato esclusivamente al software ERP. Lantek è leader globale di software per le imprese che producono componenti di lamiera, tubi e profili metallici. E' leader mondiale tra le aziende produttrici di software per l'industria metallurgica grazie alle sue soluzioni per la fatturazione, i clienti, gli uffici, la rete commerciale e tecnica, l'innovazione e la conoscenza tecnologica.
Concept non è un ERP comune. E' stato sviluppato appositamente per piccole e medie imprese specializzate nel settore della metallurgia, quindi aziende che producono componenti di lamiera, tubi e profili metallici. All'atto pratico significa che il software, oltre alle classiche caratteristiche di un ERP quali la gestione di vendite, acquisti, produzione e risorse dell'impresa, ha una serie di interfacce in grado di collegarsi al software CAM che il cliente utilizza. Il motivo per cui il software è così specifico nasce dall'esigenza del settore di entrare nell'ottica dell'*Industry 4.0.* che ha come obiettivo principale l'integrazione globale del sistema produttivo

in modo innovativo e automatizzato. Grazie alla tipologia di servizi che vengono forniti, l'azienda ha accesso a tutti i database dei clienti. I database CAM contengono sostanzialmente tutti gli articoli che vengono prodotti, i materiali con i quali vengono realizzati e i disegni degli articoli dei clienti per cui vengono effettuate le lavorazioni. Il software ERP invece è molto più completo perché, oltre ad avere tutte le informazioni contenute nel DB precedente, contiene anche tutte le informazioni relative alle gestioni principali delle attività dell'impresa. In particolare, tutto ciò che riguarda lo stato del lavoro in officina, a partire dagli ordini fino ai rilevamenti e al monitoraggio della produzione. Inoltre sono contenute tutte le informazioni riguardo le vendite e gli acquisti, quindi tutti i dati relativi ai cicli documentali (ordini, ddt, fatture, note di credito, ecc). Oltre ai documenti ci sono anche tutti i dati di gestione dei magazzini, quindi gli inventari periodici, i movimenti e la valorizzazione monetaria e non della materia stoccata. Inoltre ci sono tutte le informazioni relative agli articoli prodotti e comprati dal cliente e tutte le informazioni relative alle controparti che comprano e vendono gli articoli.

Il *Data Mining* è l'insieme di tecniche e metodologie che hanno per oggetto l'estrazione di informazioni utili da grandi quantità di dati attraverso metodi automatici o semi-automatici (es. machine learning) e l'utilizzo scientifico, aziendale/industriale o operativo delle stesse. Ho studiato queste tecniche di analisi di dati durante il corso di Business Intelligence per Big Data, durante il quale, per imparare le metodologie, abbiamo implementato queste analisi su dataset di prova. Durante il tirocinio quindi ho avuto l'idea di effettuare le operazioni di *Data Mining* utilizzando un database di un cliente cercando quindi di trovare informazioni utili su dati reali compilati dal cliente in anni di utilizzo del software ERP. Una volta deciso il tipo di progetto che volevo effettuare ho cercato di trovare il cliente adatto per il tipo di database. Il cliente su cui ho scelto di fare l'analisi è un utente con anni di esperienza sia con *Lantek* che con *Concept*. In particolare utilizza *Concept* dal 2012 e ormai da qualche anno ha un utilizzo sostanzialmente completo di tutte le funzionalità che il software fornisce. Questo significa che ha un database globalmente popolato, a partire dalle anagrafiche delle controparti, fino alla gestione dei movimenti di magazzino, di pianificazione e schedulazione delle risorse.

Le tecniche di analisi di dati implementate non hanno la presunzione di ottenere un risultato in grado da poter essere un servizio finale per il cliente, ma rappresentano il primo passo di *Data Mining* tramite il quale può essere utile capire quali scelte di business effettuare.

1. Data Selection

Il database contiene tutti i dati del software ERP quindi contiene tutti i dati del cliente che fanno riferimento alle vendite, agli acquisti, alla produzione, alla pianificazione e alla gestione magazzino dell'azienda.
È chiaro quindi che, all'interno del database è presente un'elevatissima mole di dati estremamente eterogenei tra loro. Se non ci si pongono obiettivi specifici questo tipo di progetto potrebbe potenzialmente durare all'infinito perché la ricerca di informazioni utili non finirebbe mai. La prima operazione necessaria è stata quindi quella di *data selection.*

Ho analizzato la struttura del database e ho cercato di valutare quali potessero essere le tabelle più rilevanti su cui fare l'analisi e scartare tutte le altre in modo da rendere gestibile l'elevata mole di dati. Il database è composto da qualche migliaio di tabelle suddivise tra tabelle di appoggio, tabelle di sistema, viste standard e viste personalizzate (cartelle di ricerca) in base alle richieste del cliente.

Ho selezionato sei tabelle su cui fare l'analisi. Tutte le tabelle scelte fanno riferimento agli ordini di acquisto, di vendita e di conto lavoro del cliente. Queste selezionate sono tabelle *core* perché contengono dati di testata e di dettaglio dei documenti che sono fondamentali in un software ERP.

In particolare, le prime tre tabelle fanno riferimento alla testata degli ordini. Infatti tengono traccia, sostanzialmente, di qual è la ragione sociale coinvolta nell'ordine (che sia questa cliente, fornitore di beni fisici o fornitore di prestazioni) e dell'importo totale di tutto quello che è compreso nell'ordine.

Le tabelle in questione sono:

- *A01_DOC_VEN_ORD*: Ordini di Vendita
- *A01_DOC_ACQ_ORD*: Ordini di Acquisto
- *A01_DOC_CTL_ORD*: Ordini di Conto Lavoro

Dopodiché ho scelto altre tabelle che sono molto più dettagliate per l'ordinato. Infatti, sempre in riferimento agli acquisti, vendite e Conto Lavoro, si tratta di tabelle che entrano nel merito di ogni ordine. I contenuti di questi dataset sono le righe di ogni ordine che è stato effettuato e quindi le informazioni relative agli articoli, alle quantità relative di ogni ordine, ai ricavi di ogni articolo ordinato e alle informazioni relative allo stato di avanzamento dell'ordine.

Le tabelle in questione sono:

- *A01_SRC_DOC_VER*: Tutte le righe dei documenti di Vendita
- *A01_SRC_DOC_ACR*: Tutte le righe dei documenti di Acquisto
- *A01_SRC_DOC_CTR*: Tutte le righe dei documenti di Conto Lavoro

Non ho provato tutte le tecniche su tutte le tabelle ma ho cercato di scegliere tabella per tabella la tecnica che era più adatta per il tipo di dato contenuto. Le prime tre tabelle le ho utilizzate solo per fare un'analisi visiva dei dataset, in modo da avere un'idea iniziale di quella che è la struttura e la distribuzione dei dati all'interno delle tabelle. Sulle righe degli ordini invece ho effettuato la stragrande maggioranza delle tecniche, a partire dal *clustering*, fino a un lavoro specifico di *classification* degli attributi.

Come detto precedentemente, il cliente utilizza *Concept* dal 2012, è chiaro dunque che, nonostante la selezione delle tabelle, i dati contenuti all'interno delle stesse fossero ancora molto corposi. Ho scelto quindi di fare una scrematura temporale filtrando solo i dati del 2018 e del 2019, sostanzialmente per avere un prospetto completo solo sugli ultimi due anni di lavoro.

Per eseguire le operazioni di *Data Mining* utilizzo il tool *Rapid Miner* che contiene tutti i modelli basilari da applicare su dataset di questo tipo.

Per poterlo utilizzare ho fatto le query a database in modo da poter estrarre i dati appena descritti, ho esportato i risultati in formato *.csv*, form disponibile per la lettura dati in *Rapid Miner.* Di seguito riporto le query utilizzate per estrarre i dati di interesse:

- Ordini di vendita: da 16498 a 3709 record (70,789 MB – 130 attributi)

```
SELECT *
FROM A01_DOC_VEN_ORD
WHERE (DTANN = 2019 OR DTANN = 2018)
```

- Ordini di acquisto: da 10363 a 2364 record (38,805 MB – 106 attributi)

```
SELECT *
FROM A01_DOC_ACQ_ORD
WHERE (DTANN = 2019 OR DTANN = 2018)
```

- Ordini di conto lavoro: da 12146 a 2768 record (39,586 MB – 102 attributi)

```
SELECT *
FROM A01_DOC_CTL_ORD
WHERE (DTANN = 2019 OR DTANN = 2018)
```

- Tutte le righe dei documenti di Vendita: da 399381 a 18733 record (521,742 MB – 115 attributi)

```
SELECT *
FROM A01_SRC_DOC_VER
WHERE (DTANN = 2018 OR DTANN = 2019)
AND (DTCAU = 'ORDVE' OR DTCAU = 'ORDCL')
```

- Tutte le righe dei documenti di Acquisto: da 40575 a 5603 record (203,961 MB – 105 attributi)

```
SELECT *
FROM A01_SRC_DOC_ACR
WHERE (DTANN = 2018 OR DTANN = 2019)
AND DTCAU = 'ORDAQ'
```

- Tutte le righe dei documenti di Conto Lavoro: da 294562 a 16830 record (294,789 MB – 106 attributi)

```
SELECT *
FROM A01_SRC_DOC_CTR
WHERE (DTANN = 2018 OR DTANN = 2019)
AND DTCAU = 'ORDLV'
```

Per semplicità ho riportato le query che estraggono tutti gli attributi di ogni tabella, in realtà non ho selezionato tutti i campi da esportare. Ogni tabella è composta da centinaia di *regular attributes* che sono tipicamente campi che vengono compilati durante l'utilizzo del software dal cliente. Molti di questi campi erano vuoti e quindi erano impostati al valore di default. Essendo di fatto *missing attributes*, ho effettuato una prima fase di *pre-processing* escludendo dalla selezione circa il 50% di attributi per ogni dataset considerato.

2. Analisi visiva del dataset

Nella prima parte di studio del database ho effettuato l'analisi visiva del dataset per ottenere delle prime informazioni utili senza la necessità di applicare modelli e/o algoritmi ma soltanto con un approccio di primo impatto. Tabella per tabella ho cercato grafici che mi dessero un'idea del tipo di dati da dover gestire. In questa fase non si possono ottenere informazioni nascoste ma è un primo momento importante perché dà un'idea rapida su quelle che possono essere le indicazioni estraibili dal DB. Di seguito sono riportati tutti i dettagli.

2.1 Analisi visiva Ordini di Vendita

Questa tabella, come detto in precedenza, contiene tutte le testate dei documenti di ordine di vendita. La vendita, concepita in *Concept*, si divide in due modalità distinte: la vendita di un bene fisico e la fornitura di una prestazione. Tutte le vendite sono però incluse nella stessa tabella quindi, per distinguere le due modalità, bisogna controllare l'attributo DTCAU che indica la causale del documento. In particolare, se si tratta della vendita di un bene tangibile la causale dell'ordine è "*ORDVE*", al contrario, se si tratta della fornitura di una prestazione, la causale dell'ordine è "*ORDCL*".

La tipologia di cliente che utilizza *Concept* è tipicamente un PMI specializzata nel settore metalmeccanico. Questo tipo di imprese, soprattutto nel bacino italiano, sono contoterziste. Ciò significa che non si occupano solo di produttore prodotti finiti da vendere al cliente finale ma, molto spesso il cliente stesso è il fornitore della materia prima da lavorare, quindi quello che viene venduto dal produttore non è tanto il bene fisico ma la prestazione (es. taglio, piega, saldatura) sulla materia fornita dal cliente stesso. Questo tipo di processo è detto Conto Lavoro.
Ho cercato quindi di capire com'era organizzata la struttura delle vendite del cliente preso in esame. Per farlo ho guardato come le ragioni sociali

coinvolte negli ordini del cliente erano suddivise tra vendita e Conto Lavoro. Come si nota in Figura **1** il cliente ha la principale parte di vendita imputata alla fornitura di beni fisici. La parte di Conto Lavoro comprende neanche 635 ragioni sociali rispetto alle più di 3074 che riguardano le vendite. Le ragioni sociali per le quali sono stati fatti ordini di Conto Lavoro non sono diverse da quelle per le quali sono stati fatti ordini di vendita. In sostanza tutti i clienti comprano prodotti finiti dal cliente e circa il 17% di loro chiede anche una fornitura di prestazioni. Il grafico a torta mostra quanto appena descritto, la variabile analizzata è *DTCAU* che corrisponde alla causale del documento. L'attributo in questa tabella può assumere due valori:

- *ORDCL:* se si tratta di una fornitura di prestazione
- *ORDVE:* se si tratta della vendita di un bene fisico

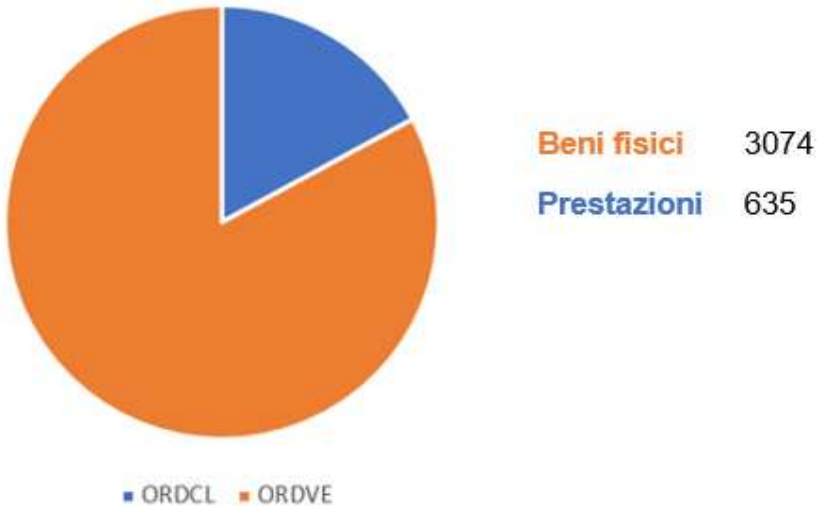

Figura 1: Rapporto Vendita – Conto Lavoro

In seguito a questa prima suddivisione ho cercato di capire se ci fosse qualche informazione utile riguardante gli importi dei documenti. Ho visualizzato un altro grafico dove, sulle ascisse sono presenti le ragioni sociali mentre sulle ordinate la somma dei totali documento espressi in €. L'obiettivo era quello di verificare se ci fossero ragioni sociali a cui fossero imputati importi totali rispetto ad altre. Il grafico in Figura **2** indica chiaramente come i totali siano omogeneamente distribuiti per quasi tutte le ragioni sociali ad eccezione di una che ha un importo nettamente superiori a tutte le altre.

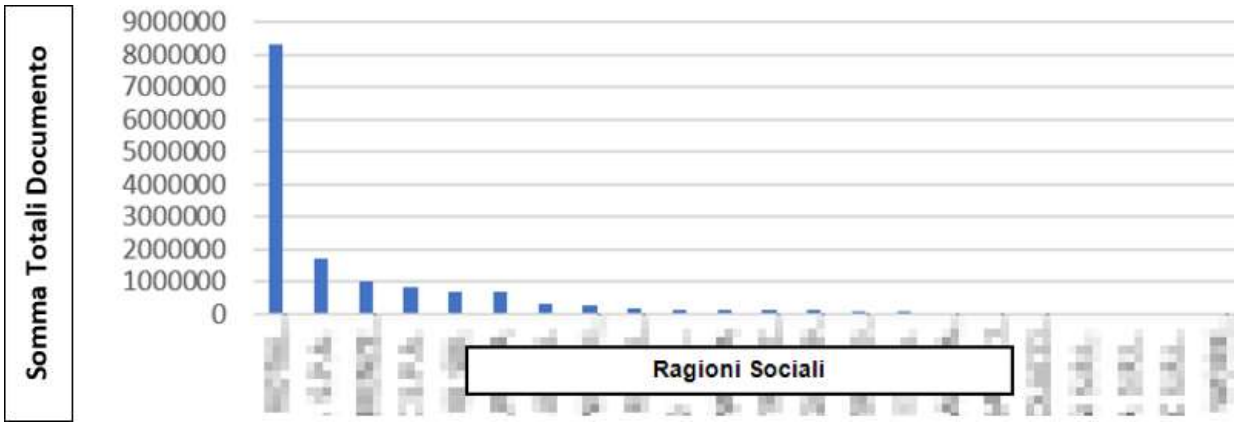

Figura 2: Scatter Ragioni Sociali – Somma dei totali documento

Un’altra statistica che ho verificato è quella di vedere lo scenario geografico di dove si trovano i clienti al quale vengono fatti gli ordini di vendita. Come si può notare in Figura **3** la stragrande maggioranza dei clienti ha sede a Scarperia, nella stessa città del cliente. In ogni caso la quasi totalità dei clienti è in Provincia di Firenze e quindi nel bacino di utenza del cliente.

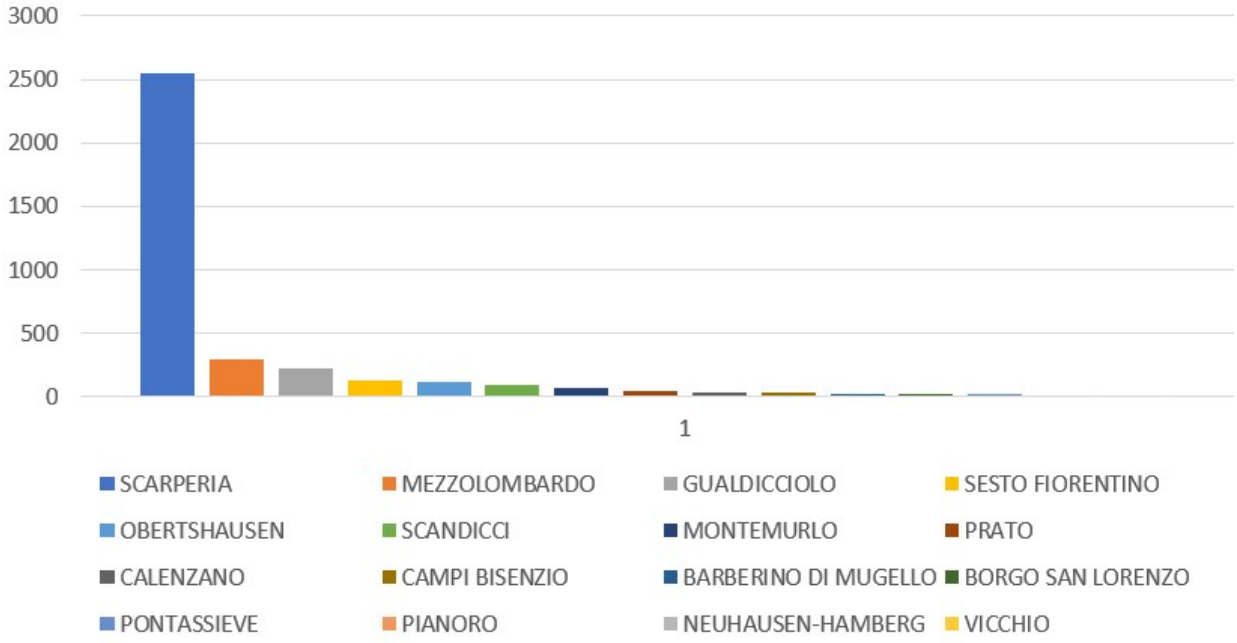

Figura 3: Scatter Città – Contatore Ragioni Sociali

2.2 Analisi visiva Ordini di Conto Lavoro

La tabella di Conto Lavoro tiene in considerazione tutti gli ordini di Conto Lavoro che il cliente fa alle controparti. Questo tipo di processo in realtà è detto Conto Lavoro passivo. Gli ordini finiscono in questa tabella quando il cliente che utilizza *Concept* chiede la fornitura di una prestazione alla controparte ma il materiale che viene lavorato è di sua proprietà e non della controparte stessa. In questo caso si tratta di bene intangibile di conseguenza il ciclo su cui viene gestito é diverso dall'acquisto classico di articoli dai fornitori.

Come statistica ho cercato di capire geograficamente dove si trovavano i fornitori di Conto Lavoro ai quali il cliente fa le maggiori richieste. La Figura **4** indica la distribuzione geografica delle città in relazione alle ragioni sociali coinvolte. Come si può notare tutti i fornitori di prestazioni sono situati a Scarperia, la stessa città del cliente. San Piero è una frazione della città stessa. Questo dato, rispetto a quello delle vendite, non stupisce perché il Conto Lavoro viene usato tipicamente quando devono essere fatte lavorazioni particolari che il cliente non è in grado di fare ma soprattutto quando il cliente, per ragioni di tempo, non è in grado di completare tutte le lavorazioni per le date prestabilite.

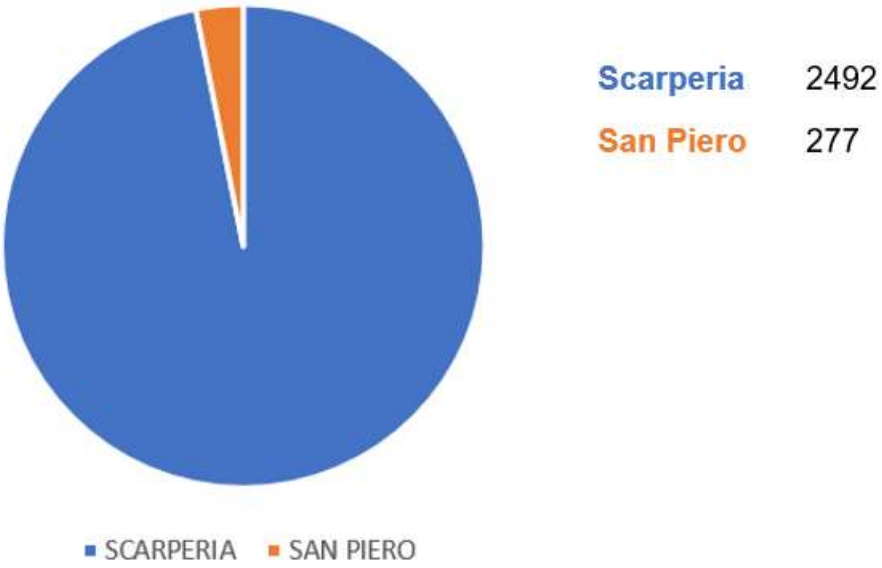

Figura 4: Città dei fornitori di Conto Lavoro

2.3 Analisi visiva Ordini di Acquisto

La tabella degli acquisti riguarda tutte le testate dei documenti degli acquisti di materiale che fa il cliente. In questo caso la statistica che ho cercato è stata quella di capire come venivano distribuiti i totali dei documenti in base alle ragioni sociali. Come si può notare in Figura **5** alla maggior parte dei fornitori corrispondono totali documento inferiori ai 25/30 k€, mentre per una decina circa di fornitori, i totali sono molto più elevati.

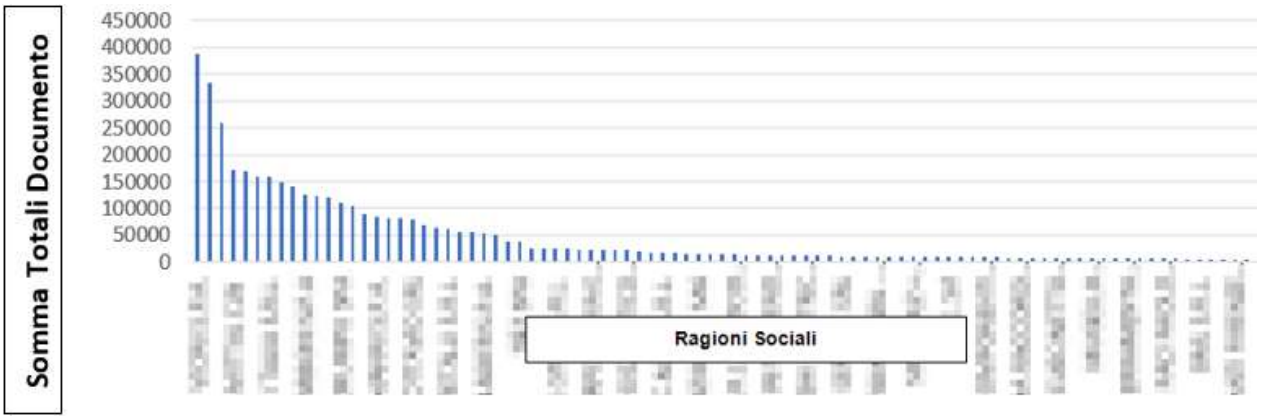

Figura 5: Scatter Ragioni Sociali – Somma Totali Documento

3. Pre-processing

Dopo la fase di *data selection* ho approfondito le operazioni di analisi del database.

La prima tecnica da applicare obbligatoriamente in ambito di *Data Mining* è il *pre-processing.* I dati di un database, in una realtà aziendale, sono tipicamente *"sporchi"*. I dati vengono compilati dal cliente attraverso l'utilizzo quotidiano di *Concept*, quindi molto frequentemente si trovano dati incompleti, cioè che ci siano attributi non valorizzati oppure che ci sia una mancanza di attributi per gli scopi di analisi. Questi dati inoltre possono essere *"rumorosi"*, nel senso che contengono errori oppure che ci sia la presenza di *outliers*, cioè valori anomali che chiaramente si distinguono dal resto delle osservazioni disponibili. Un altro problema che può essere gestito con il *preprocessing* è il caso di dati inconsistenti, cioè dati che hanno discrepanze nei nomi oppure nei codici degli attributi. Il motivo principale per cui si effettua *preprocessing* si basa sulla *legge GIGO: "garbage in, garbage out"*. In ambito informatico con questo enunciato si fa riferimento al fatto che nessun algoritmo può portare a risultati soddisfacenti se le risorse in input non sono di qualità. Anche in Data Mining, quindi, se i dati in input sono di bassa qualità, i risultati nel processo di mining saranno poco consistenti.

I problemi principali riguardo al database preso in esame sono l'elevata presenza di dati mancanti poiché non sono stati compilati dal cliente durante l'uso di *Concept*. Di per sé non è un errore se il cliente non compila tutti i dati. Le tabelle del DB sono strutturate in modo tale per cui siano presenti tutti gli attributi che garantiscano un utilizzo completo e integrale del software. Non è detto che al cliente possano interessare effettivamente tutte le funzionalità che sono garantite, quindi è molto probabile che certi campi non vengano mai utilizzati perché futili per la gestione produttiva e organizzativa dell'azienda.

Procedendo per gradi ho quindi esplorato una per una le tabelle scelte precedentemente e, per ognuna di loro, attraverso l'utilizzo di operatori di *RapiMiner*, ho cercato di filtrare e sostituire gli attributi mancanti. Operativamente ho agito secondo la seguente sequenza:

1. Il primo passo è stata la lettura del file *.csv*. In questa fase è necessario verificare, tramite la fase di configurazione di import, che tutti gli attributi abbiano un'unità di misura coerente con il contenuto effettivo dei dati. Per ognuno degli attributi ho quindi definito la classe tra *polynomial*, *binomial*, *numerical*, *real*, *date-time* (coerentemente con il formato del DB).

2. Dopo la prima lettura ho controllato quali fossero le colonne interamente vuote e tramite l'operatore *Select Attributes* ho filtrato gli attributi in modo da escluderli dai passi successivi dell'analisi. Inoltre ho escluso dati che non erano mancanti ma che erano irrilevanti ai fini dell'analisi, ad esempio campi uguali per tutte le tuple che non dessero quindi informazioni utili aggiuntive.

3. Dopodiché ho cercato i valori mancanti e li ho sostituiti con il valore medio calcolato in base a tutti i valori presenti nelle altre osservazioni in corrispondenza di quel determinato attributo. Ovviamente se si è trattato di dati numerici il calcolo è stato più immediato, nel caso di altri tipi di dati come ad esempio quelli testuali, il valore mancante è stato sostituito dal valore più frequente di quell'attributo. Ad esempio per quanto riguarda la modalità di pagamento, dove non era specificata è stata inserita la "*BB120*", cioè il bonifico bancario di 120 giorni che è la modalità più ricorrente. Gli operatori che ho utilizzato sono *Declare Missing Values* e *Replace Missing Values.*

4. L'ultimo passo operativo che ho effettuato è stato quello di filtrare ulteriormente i dati nel caso in cui fossero rimasti ancora dati mancanti con l'operatore *Filter Examples*.

In Figura **6** è rappresentato il processo utilizzato in *Rapid Miner*.

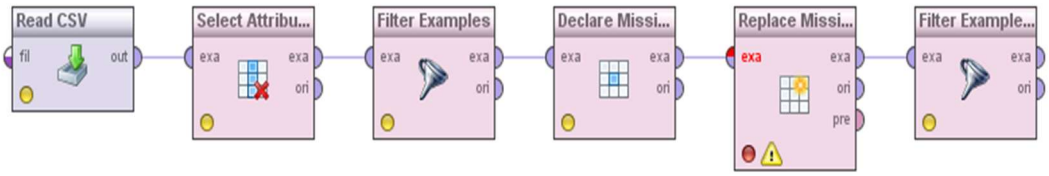

Figura 6: Preprocessing Rapid Miner

Due fasi importanti del *preprocessing* sono la parte di *data reduction* e di *discretization.* In questa prima parte del processo ho saltato la parte di discretizzazione perché non ho ritenuto rilevante al momento concentrarmi su questo aspetto e non ho effettuato un campionamento dei dati perché ho deciso di pensare successivamente al tipo di campionamento adatto in base all'analisi che mi trovavo a fare in quel momento.

Di seguito riporto tutti i dati di preprocessing relativi alle singole tabelle.

3.1 Preprocessing Ordini di Vendita

- 3709 osservazioni prima *preprocessing*
- Attributi mancanti iniziali: 941 elementi
- Filtro di attributi mancanti:
  - DTBAN: *missing attribute*
  - DTCCO: *missing attribute*
  - DTCIN: *missing attribute*
  - DTIBA: *missing attribute*
  - DTFIS: *missing attribute*
  - DTPIV: *missing attribute*
  - DTTPD: tipo documento 'ORD' per tutte le tuple
  - DTRSS: ragione sociale di spedizione, sempre uguale a DTRAS
  - DTRIC: riferimento alla controparte.

Sufficiente DTRAS

    - DTPER: riferimento alla controparte. Sufficiente DTRAS
- Attributi mancanti dopo il filtro: 179 elementi (4.8%)
- Dopo aver sostituito i dati mancanti ho ulteriormente rimosso 7 righe
- 3702 osservazioni dopo *preprocessing*

3.2 Preprocessing Ordini di Acquisto

- 2364 osservazioni prima *preprocessing*
- Attributi mancanti iniziali: 1742 elementi
- Filtro di attributi mancanti:
    - DTABI: *missing attribute*
    - DTCAB: *missing attribute*
    - DTCAY: *missing attribute*
    - DTLIS: *missing attribute*
    - DTNOS: *missing attribute*
    - DTPOR: *missing attribute*
    - DTRFI: *missing attribute*
    - DTPER: riferimento alla controparte. Sufficiente DTRAS
    - DTFIS: riferimento telefono fisso controparte fornitore
    - DTPIV: riferimento cellulare controparte fornitore
- Attributi mancanti dopo il filtro: 61 elementi (2.6%)
- Dopo aver sostituito i dati mancanti ho ulteriormente rimosso 4 righe
- 2360 osservazioni dopo *preprocessing*

3.3 Preprocessing Ordini di Conto Lavoro

- 2768 osservazioni prima *preprocessing*
- Attributi mancanti iniziali: 2458 elementi

- Filtro di attributi mancanti:
    - DTABI: *missing attribute*
    - DTAEB: *missing attribute*
    - DTBAN: *missing attribute*
    - DTCAB: *missing attribute*
    - DTCAP: *missing attribute*
    - DTCCO: *missing attribute*
    - DTCIN: *missing attribute*
    - DTCIT: *missing attribute*
    - DTIBA: *missing attribute*
    - DTNRC: *missing attribute*
    - DTPRO: *missing attribute*
    - DTRSS: *missing attribute*
    - DTVIA : *missing attribute*
    - DTRFI: *missing attribute*
    - DTFIS: riferimento telefono fisso controparte fornitore
    - DTPIV: riferimento cellulare controparte fornitore
    - USINS,DTINS,USMDF,DTMDF: utente e date di inserimento e modifica
- Attributi mancanti dopo il filtro: 105 elementi (3.8%)
- Dopo aver sostituito i dati mancanti ho ulteriormente rimosso 31 righe
- 2737 osservazioni dopo *preprocessing*

3.4 Preprocessing righe degli ordini di Conto Lavoro

- 16830 osservazioni prima *preprocessing*
- Attributi mancanti iniziali: 4815 elementi
- Filtro attributi mancanti:
    - DRCMM: *missing attribute*
    - DTMDF,USMDF,DTINS,USINS: utente e date di inserimento e modifica
    -

- Attributi mancanti dopo il filtro: 146 elementi (3.1%)
- 16830 osservazioni dopo *preprocessing*

3.5 Preprocessing righe degli ordini di Vendita

- 18733 osservazioni prima *preprocessing*
- Attributi mancanti iniziali: 9071 elementi
- Filtro attributi mancanti:
    - DRCMM: *missing attribute*
    - ARMAT: *missing attribute*
    - DTPER: riferimento alla controparte. Sufficiente DTRAS
    - DTMDF,USMDF,DTINS,USINS: utente e date di inserimento e modifica
- Attributi mancanti dopo il filtro: 224 elementi (1.2%)
- 18509 osservazioni dopo *preprocessing*

3.6 Preprocessing righe degli ordini di Acquisto

- 5602 osservazioni prima *preprocessing*
- Attributi mancanti iniziali: 4424 elementi
- Filtro attributi mancanti:
    - DRCMM: *missing attribute*
    - ARMAT: *missing attribute*
    - DTPER: riferimento alla controparte. Sufficiente DTRAS
    - DTMDF,USMDF,DTINS,USINS: utente e date di inserimento e modifica
- Attributi mancanti dopo il filtro: 138 elementi (2.4%)
- 5464 osservazioni dopo *preprocessing*

4. Clustering

Clustering è un tipo di tecnica utile a capire come sono strutturati i dati all'interno di una tabella. Con *Clustering* si intendono l'insieme di tecniche grazie alle quali effettuare raggruppamenti degli elementi. L'obiettivo del *Clustering* è quello di segmentare i dati, ovvero di stabilire l'esistenza di gruppi/classi omogenei nei dati*[1]*.

4.1 Clustering dati strutturati

Il primo tipo di *Clustering* che ho testato è quello dei dati strutturati. Le tabelle su cui ho effettuato gli algoritmi sono quelle che contengono le righe degli ordini. Questo tipo di tabelle contengono dati semi-strutturati, poiché oltre ad avere attributi strutturati numerici, hanno campi testuali come le descrizioni degli articoli e le note degli ordini.

L'algoritmo su cui ho basato le mie analisi è il *K-Means*. Per questo tipo di algoritmo è necessario settare il parametro *k* che indica il numero di cluster in cui vengono suddivisi i dati. Questo parametro non può essere selezionato in modo casuale ma tramite l'applicazione di alcune euristiche può essere trovato il parametro ottimale da utilizzare.

Il metodo più corretto da utilizzare sarebbe quello del *SSE[2].* Questo metodo è analitico, tramite l'operatore *Execute Script*, variando il valore di *k*, si seleziona il parametro a cui corrisponde il valore di *SSE*, scarto quadratico minore, presente nel file di log dopo il lancio dell'algoritmo. Questo metodo è senz'altro il più preciso ma ha lo svantaggio di avere tempi di calcolo molto lunghi, soprattutto se la dimensioni di dati su cui fare i calcoli è elevata.

Un metodo alternativo è quello delle *k-distances[3]*. Questo metodo è più di natura euristica quindi meno analitico rispetto al precedente, consiste nel lanciare l'algoritmo e. tramite l'operatore *Data to Similarity*, verificare dal grafico delle *k-distances* qual è il parametro *k* che minimizza la presenza di dati non correttamente suddivisi. In Figura **7** è presente un

esempio di grafico, i dati a sinistra della curva sono quelli che, per quel determinato valore di *k*, non vengono clusterizzati nel modo corretto. Si sceglie il *k* quindi che mostra un grafico con il minor numero di dati possibili alla sinistra della curva di distanza. In Figura **8**, **9**, **10** sono mostrati i processi *Rapid Miner* citati in precedenza. Non sono in grado di valutare la correttezza dei cluster dei dati strutturati. Questo perché non sono in grado di verificare in quanti cluster devono essere effettivamente segmentati i dati e, inoltre, non so esattamente come i cluster debbano essere popolati. Tuttavia, guardando le dimensioni dei cluster, si può dire che il modello di Clustering di SRC_DOC_VER e di SRC_DOC_CTR è più ragionevole rispetto al modello di Clustering SRC_DOC_ACR, questo perché i cluster di questi due dataset sono popolati in maniera più omogenea rispetto al terzo. Di seguito sono riportati i dettagli dei modelli dei dataset.

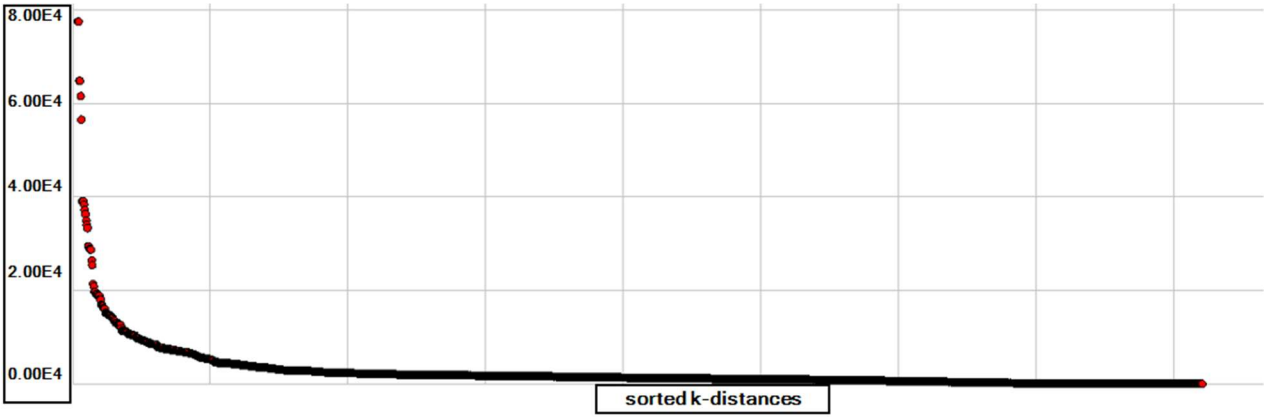

Figura 7: Grafico delle k-distances

Figura 8: Processo SSE

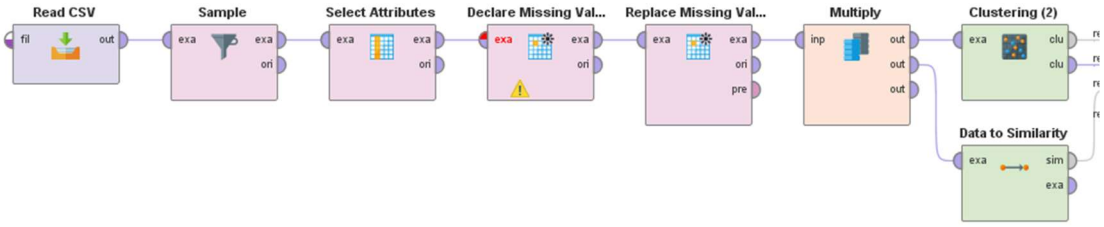

Figura 9: Processo k-distances

Figura 10: Processo di cluster

4.1.1 Clustering dati strutturati Ordini di Acquisto

L'algoritmo utilizzato è un **K-Means***[4]* con **k=14**

Cluster	#items
0	850
1	341
2	455
3	131
4	75
5	131
6	1212
7	20
8	21
9	981
10	516
11	108
12	404
13	231
Totali	**5476**

Tabella a: Cluster Model

Come detto in precedenza questo modello di Clustering potrebbe non essere ottimale perché le dimensioni dei cluster variano molto tra di loro.

4.1.2 Clustering dati strutturati Ordini di Vendita

L'algoritmo utilizzato è un **K-Means** con **k=12**

Cluster	#items
0	978
1	1070
2	942
3	37
4	593
5	891
6	1103
7	1180
8	783
9	651
10	1101
11	559
Totali	**9888**

Tabella b: Cluster Model

Questo modello appare più coerente rispetto al dataset acquisti in quanto i dati sono omogeneamente distribuiti all'interno dei cluster.

4.1.3 Clustering dati strutturati Ordini di Acquisto

L'algoritmo utilizzato è un **K-Means** con **k=12**

Cluster	#items
0	415
1	777
2	943
3	21
4	965
5	889
6	674
7	980
8	1031
9	797
10	614
11	809
Totali	**8915**

Tabella c: Cluster Model

Questo modello appare più coerente rispetto al dataset acquisti in quanto i dati sono omogeneamente distribuiti all'interno dei cluster.

4.2 Clustering di testo

Il secondo tipo di clustering che ho provato a effettuare è stato quello testuale. Di ogni tabella ho selezionato i campi testuali più rilevanti in modo da ottenere informazioni utili relative ai record e eventualmente trovare regole di associazione tra parole ricorrenti all'interno dei cluster trovati. Ho processato gli attributi testuali tramite l'operatore *Process Document from Data* in modo da ottenere la matrice delle radici più ricorrenti all'interno dei testi. Una volta processato il testo ho applicato gli algoritmi di K-Means settando come parametri i k trovati in precedenza basandosi empiricamente sui dati delle *k-distances*[3]. Tramite i sotto processi, ho applicato un sotto processo per ogni cluster trovato in modo da trovare regole di associazione interne. Sostanzialmente ho cercato di capire come venivano raggruppati gli elementi in base agli *itemset* frequenti[5] e alle regole d'associazione[6] trovate all'interno del cluster.

Process

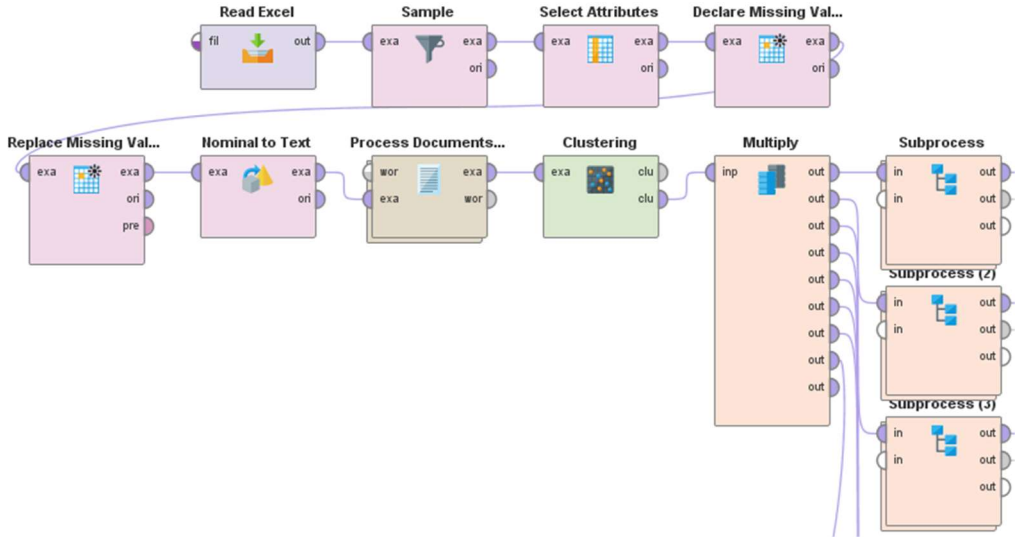

Figura 11: Process Rapid Miner

Process Document

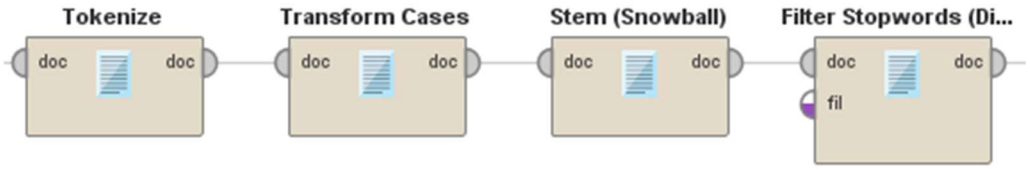

Figura 12: Process Document From Data

Subprocess

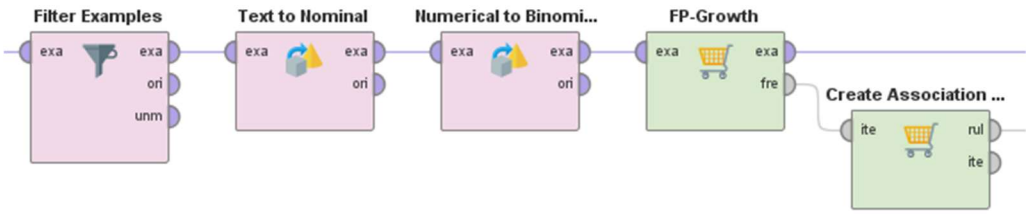

Figura 13: Sub-Process Rapid Miner

4.2.1 Cluster testuale Ordini di Vendita

Per questa tabella gli attributi che ho preso in considerazione sono DRART e DRDES che corrispondono rispettivamente al codice e alla descrizione dell'articolo. Questo tipo di campi tipicamente contengono in modo riassuntivo i dati principali relativi agli articoli quali il materiale utilizzato, lo spessore e il tipo di funzionalità che ha.

L'algoritmo di clustering utilizzato è un K-Means con k=12, in Tabella **d** è rappresentata la suddivisione dei record all'interno dei cluster. Per problemi con la licenza gratuita nell'utilizzo di *Rapid Miner 9* ho effettuato un campionamento del 53%.

Cluster	#items
0	98
1	2143
2	606
3	507
4	431
5	213
6	4284
7	338
8	278
9	212
10	546
11	272
Totali	**9928**

Tabella d: Cluster Model

Dopodiché ho ricercato eventuali regole di associazione tra *itemset* frequenti*[5]*. Di seguito sono rappresentati alcuni esempi di regole trovati per ogni cluster. Tra parentesi è indicato il parametro di confidenza della regola. Tutte le regole in figura sono rilevanti poiché ho controllato il parametro *lift[6]* ed era in un intervallo tra 1.58 e 3.76, quindi sempre maggiore di 1.

Come si può notare, le associazioni trovate fanno riferimento a dati sugli articoli, ad esempio tavoli e supporti cernita in acciaio inox, oppure tubazioni anti condensa, o componenti per bracci rotatori automatici. Come si nota non sono tutte radici di parole di senso compiuto ma non è una cosa che stupisce perché sono stati processati anche i codici degli

articoli che quindi sono sigle alfanumeriche che hanno una codifica precisa per il cliente al momento dell'inserimento in *Concept.*

Cluster 0:

```
[tavol] --> [cern, aa] (confidence: 0.753)
[cern, l] --> [d, inox] (confidence: 0.816)
[cern, support, l] --> [inox] (confidence: 0.986)
[fasc] --> [esa, inox] (confidence: 0.853)
[inox, fasc] --> [bc, x, esa] (confidence: 0.856)
```

Cluster 3:

```
[tubaz, cal, vapor] --> [oh] (confidence: 0.714)
[tubaz, oh] --> [cal] (confidence: 0.736)
[collettor] --> [tubaz, d, condens] (confidence: 0.737)
[tubaz, d, h] --> [condens] (confidence: 0.756)
```

Cluster 4:

```
[std] --> [r, rotary] (confidence: 0.718)
[h, std] --> [r, b, rotary] (confidence: 0.730)
[prodott] --> [b, rotary] (confidence: 0.978)
[colonn] --> [b, rotary] (confidence: 0.978)
```

Cluster 8:

```
[oh, carter] --> [sott] (confidence: 0.700)
```

4.2.2 Cluster testuale Ordini di Conto Lavoro

Per questa tabella invece, oltre agli attributi precedenti, ho selezionato anche gli attributi testuali relativi alla lavorazione per il quale l'articolo viene mandato in Conto Lavoro, in particolare DRLAV, nome della lavorazione e DRLDS, codice della lavorazione.

L'algoritmo di clustering utilizzato è un K-Means[4] con k=12, in Tabella **e** è rappresentata la suddivisione dei record all'interno dei cluster. Per problemi con la licenza gratuita nell'utilizzo di *Rapid Miner 9* ho effettuato un campionamento del 59%.

Cluster	#items
0	480
1	413
2	281
3	5338
4	258
5	258
6	366
7	300
8	1354
9	195
10	243
11	443
Totali	**9928**

Tabella e: Cluster Model

Dopodiché ho ricercato eventuali regole di associazione tra *itemset* frequenti. In figura sottostante sono rappresentati alcuni esempi di regole trovati. Tutte le regole in figura sono rilevanti poiché ho controllato il parametro *lift* ed era in un intervallo tra 5.23 e 7.62, quindi sempre maggiore di 1. Come si può notare ci sono molti riferimenti alle lavorazioni esterne come l'imballaggio e la verniciatura di strutture probabilmente di dimensioni molto elevate che il cliente non è in grado di effettuare internamente e affida a un'impresa esterna più specializzata in quel tipo di lavorazione.

Cluster 0

```
[estern] --> [cern] (confidence: 0.736)
[imball] --> [estern, cern] (confidence: 0.736)
[imball, ime] --> [estern, d, cern] (confidence: 0.736)
```

Cluster 2:

```
[ver, x] --> [h, verniciatur] (confidence: 0.981)
[ver, x] --> [h, o, verniciatur] (confidence: 0.981)
```

Cluster 3:

```
[struttur] --> [verniciatur, h] (confidence: 0.742)
[pianton, struttur] --> [ver, h] (confidence: 0.742)
[c, struttur] --> [pianton, ver, verniciatur, h, o] (confidence: 0.742)
[verniciatur, struttur] --> [pianton, h] (confidence: 0.748)
```

Cluster 7:

```
[estern] --> [aa, laser] (confidence: 0.744)
[estern] --> [aa, tagl] (confidence: 0.744)
[tub, estern] --> [c, aa, tagl] (confidence: 0.744)
[estern] --> [toe, tornitur] (confidence: 0.786)
```

Cluster 9:

```
[quertrager] --> [verniciatur] (confidence: 1.000)
```

4.2.3 Cluster testuale Ordini di Acquisto

Per la tabella acquisti ho considerato gli stessi campi delle vendite di beni fisici quindi il codice e la descrizione dell'articolo.

L'algoritmo di clustering utilizzato è un K-Means con k=14, in Tabella **f** è rappresentata la suddivisione dei record all'interno dei cluster.

Cluster	#items
0	214
1	242
2	140
3	157
4	18
5	2510
6	522
7	386
8	134
9	627
10	282
11	147
12	185
13	39
Totali	**5603**

Tabella f: Cluster Model

Dopodiché ho ricercato eventuali regole di associazione tra *itemset* frequenti. In figura sottostante sono rappresentati alcuni esempi di regole trovati. Tutte le regole in figura sono rilevanti poiché ho controllato il parametro *lift* ed era in un intervallo tra 1.52 e 2.15, quindi sempre maggiore di 1.

Come prevedibile, ci sono chiari riferimenti all'acquisto di lamiere e ai particolari come viti o tubolari che devono essere realizzati.

Cluster 0:

```
[est, tub] --> [bracc] (confidence: 0.899)
[tub] --> [est, bracc, per] (confidence: 0.899)
[est] --> [lamier, o] (confidence: 0.909)
```

Cluster 1:

```
[trav] --> [c, est, h] (confidence: 0.909)
[lamier] --> [c, trav, o] (confidence: 0.909)
```

Cluster 4:

```
[tagl, disegn] --> [sp, mm] (confidence: 0.703)
[m, uni, x] --> [vi, zinc, vit] (confidence: 0.707)
[uni, vit] --> [vi, m, zinc, x, feznvun] (confidence: 0.712)
[zinc, uni] --> [vi, m, vit, feznvun] (confidence: 0.719)
```

Cluster 8:

```
[consegn] --> [stab] (confidence: 0.949)
[consegn, press] --> [stab] (confidence: 0.949)
[scarper] --> [san, stab] (confidence: 0.949)
[consegn] --> [pier, press, stab] (confidence: 0.949)
```

Cluster 12:

```
[particol] --> [lam] (confidence: 0.717)
[lam, particol] --> [per] (confidence: 1.000)
```

5. Association Rules

Dopo le operazioni di *clustering* sono passato alla fase di *association rules*. L'obiettivo di quest'operazione di *mining* è quello di fare un'estrazione di correlazioni partendo da un database transazionale*[5]*.

Per effettuare queste operazioni ho considerato le tabelle delle righe degli ordini. Per utilizzare questa tecnica di analisi mi sono basato sul modello del "basket di prodotti". Questo è un classico esempio che viene utilizzato in ambito di regole associative. Supponiamo di avere una serie di scontrini dove sono elencati gli articoli acquistati. Per trovare le regole di associazione tra i prodotti bisogna come primo passo trovare tutti gli *itemset* frequenti. Attraverso quindi appositi algoritmi è possibile trovare quali e quante sequenze di articoli esistono tra i vari scontrini considerati. Una volta trovato le frequenze, tramite altri algoritmi, è possibile trovare le eventuali associazioni tra gli *itemset* frequenti.

Nel caso del database in esame ho considerato come basket di prodotti tutti gli articoli con una relazione con almeno una ragione sociale all'interno del DB. Una transazione di quelle create rappresenta quindi quello che ogni ragione sociale ha acquistato, venduto o lavorato per il cliente che utilizza *Concept*. Per creare i file sorgenti che contengono le transazioni ho fatto le query a database, ho esportato i risultati in *.csv* per poterli trattare con *Microsoft Excel*. Una volta esportati ho creato quindi una matrice per ogni tabella in modo che contenesse sulle righe le ragioni sociali più rilevanti nella tabella (quelle più richiamate) e sulle colonne tutti gli articoli di quella tabella. Le matrici possono contenere due valori, *"true"* se la relazione tra la ragione sociale e l'articolo è positiva, *"false"* altrimenti.

Chiaramente le matrici così descritte rappresentano una mole di dati estremamente elevata quindi, caso per caso, sono andato a eliminare i dati meno rilevanti ricorrendo eventualmente anche al campionamento.

In tabella **g** è riportato un esempio di matrice utilizzato

	Item 1	**Item 2**	**Item 3**
DTRAS 1	true	false	true
DTRAS 2	false	true	false
DTRAS 3	true	true	false

Tabella g: Matrice d'esempio

In figura **14** è riportato la sequenza di operatori utilizzati in *Rapid Miner*. Come si può notare l'algoritmo utilizzato per la ricerca degli *itemset* frequenti è *FP-Growth[6]*, uno degli algoritmi più utilizzati nel *Data Mining* per questa funzione.

Figura 14: Processo RapidMiner

5.1 Association Rules Ordini di Vendita

Il file sorgente coinvolto in questo test riguarda gli ordini di vendita. Ciò significa che il valore all'interno di ogni cella della matrice è *"true"* se il cliente corrispondente ha acquistato almeno una volta l'articolo.

La matrice aveva una mole troppo elevata di dati quindi ho dovuto effettuare un campionamento per riuscire a ottenere risultati in tempi ragionevoli. Il campionamento più grande che sono riuscito ad utilizzare è stato del 25%. La scelta del campionamento è stata un *random sample*. Ho usato il campionamento randomico e non stratificato perché nel secondo caso avrebbe preso ogni partizione (stratificazione) quindi sarebbe molto complicato trovare le relazioni tra le transazioni.

Itemset frequenti

Il risultato ottenuto dall'algoritmo *FP-Growth* ha portato a risultato 7 *itemset* frequenti (il dettaglio delle dimensioni in tabella **h**). In generale l'estrazione non è molto soddisfacente, ciò probabilmente è dovuto al massiccio campionamento che è stato necessario fare per permettere il funzionamento del modello.

Size itemset	#itemset
1	3
2	3
3	1

Tabella h: Tabella itemset frequenti

Regole d'associazione

Come prevedibile, le associazioni trovate non hanno dato grossi risultati significativi. Gli articoli coinvolti in regole trovate sono solamente tre:

- 0640-0081: Piastra Bloccaggio Forcella x MDH – Prodotto Finito
- 0640-0083: Supporto Frontalino x MDH – Prodotto Finito
- 3000394: Protezione Interna MNS/MDM V110 – Prodotto Finito

In figura **15** è rappresentato un grafico di regole di associazione trovate.

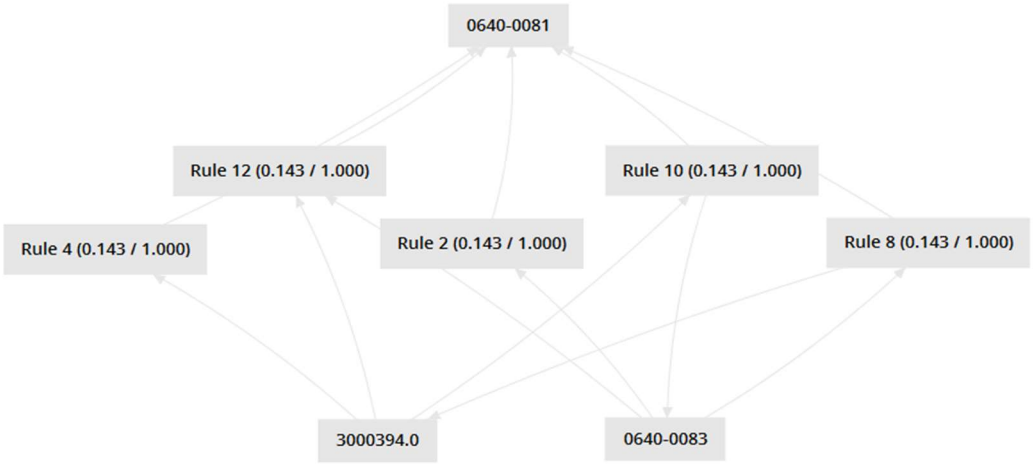

Figura 15: Esempio grafico di association rules

In tabella **i** ho inserito un esempio tabellare delle regole trovate.

Premises	Conclusion	Support	Confidence	Lift
0640-0083	3000394.0	0.143	1.0	7.0
3000394.0	0640-0083	0.143	1.0	7.0
0640-0081	0640-0083, 3000394.0	0.143	1.0	7.0
0640-0083	0640-0081, 3000394.0	0.143	1.0	7.0

Tabella i: Esempio di regole d'associazione

5.2 Association Rules Ordini di Acquisto

Il file sorgente coinvolto in questo test riguarda gli ordini di acquisto. In questo caso, quindi, il valore all'interno di ogni cella della matrice è *"true"* se l'articolo coinvolto è stato comprato almeno una volta da quel fornitore.

Come per il caso delle vendite, la matrice è risultata troppo grossa per essere gestita interamente. Al contrario di prima, però, ho deciso di non fare campionamento dati i risultati poco convincenti ottenuti. Quindi ho cercato di "sfoltire" in modo logico la matrice cercando di mantenere informazioni rilevanti senza sfalsare i dati. Per questo motivo ho scelto di includere nell'analisi solo una categoria di articoli per capire eventualmente quali relazioni ci fossero tra argomenti dello stesso tipo.

Itemset frequenti

Nella configurazione dell'algoritmo è necessario settare il parametro di supporto minimo che permette di definire un *itemset* frequente.

Il parametro di supporto indica quante volte un *itemset* frequente è presente all'interno del totale delle transazioni:

$$\sup = \frac{\#\{A, B\}}{|T|}$$

Il supporto*[7]* dell'*itemset* {A,B} corrisponde al rapporto tra il numero di volte in cui compare e il totale delle transazioni considerate. Il supporto minimo che ho considerato per queste transazioni è 0.04 in modo quindi

che vengano inclusi tutti gli *itemset* che appaiano almeno in due transazioni ed escludendo quelli che appaiono solo una volta.

Il risultato è stato un totale di 1940 *itemset* frequenti presenti in almeno due transazioni. In tabella **I** è specificato il dettaglio della distribuzione degli *itemset* in base alla dimensione, quindi al numero di articoli contenuti.

Size itemset	#itemset
1	29
2	93
3	169
4	297
5	463
6	361
7	318
8	151
9	48
10	11

Tabella I: Tabella itemset frequenti

Regole d'associazione

Dei 238 articoli, 23 sono quelli all'interno di un *itemset* per il quale è stata trovata almeno una regola di associazione. Avendo selezionato solo una tipologia di articoli le regole riguardano articoli simili. Si tratta di componentistica, nello specifico di viti con materiali, filettatura e diametro differenti. E' naturale trovare regole di associazione tra articoli di questo tipo perché sono componenti che probabilmente servono per la realizzazione di un prodotto finito composto da semilavorati differenti.

- *VI-FEZNDUNI5588-M16*
- *VI-FEZNDUNI5588-M12*
- *VI-FEZNDUNI5588-M10*
- *VI-FEZNDUNI5588-M20*
- *VI-FEZNRDIN125A-12*
- *VI-FEZNRDIN125A-16*
- *VI-FEZNVISO7380-M6X20*
- *VI-FEZNVUNI5739-M12X50*
- *VI-FEZNVUNI5931-M6X20*
- *VI-FEBRGUNI5929-M6X10*
- *VI-FEZNDUNI5588-M4*
- *VI-FEZNDUNI5588-M6*
- *VI-FEZNDUNI7473-M6*
- *VI-FEZNGUNI5929-M6X12*
- *VI-FEZNGUNI5929-M6X6*
- *VI-FEZNRDIN125A-4*
- *VI-FEZNRDIN125A-6*
- *VI-FEZNRUNI6593-18-D54*
- *VI-FEZNVISO7380-2-M8X16*
- *VI-FEZNVUNI5739-M16X55*
- *VI-FEZNVUNI5739-M16X60*
- *VI-FEZNVUNI5931-M6X16*
- *VI-FEZNVUNI5931-M8X20*

La maggior parte dei grafici sono incomprensibili perché le associazioni trovate sono tantissime. In figura **16** e **17** ho inserito alcuni esempi grafici di regole: tra parentesi sono indicati i valori di supporto e confidenza che caratterizzato la regola trovata.

Il parametro confidenza*[7]* rappresenta la frequenza di un *item* all'interno degli *itemset* in cui presente un altro *item*:

$$conf = \frac{\sup(A, B)}{\sup(A)}$$

La confidenza dell'item B con A rappresenta la probabilità di trovare B in un itemset in cui è presente A. Dopo aver trovato le regole ho cercato di capire se queste fossero effettivamente significative. Per farlo ho controllato il parametro *lift.*

Il *lift* si calcola come la correlazione statistica e corrisponde all'interesse quindi al tipo di correlazione tra gli *itemset* frequenti per i quali è stata trovata una regola di associazione:

$$corr = \frac{P(AUB)}{P(A)P(B)} = \frac{P(B|A)}{P(B)}$$

- Se corr = 1 i due eventi sono indipendenti
- Se corr > 1 i due eventi sono positivamente correlati
- Se corr < 1 i due eventi sono negativamente correlati

Si noti come, quando si tratta di regola associativa, il parametro *lift* corrisponde al rapporto tra confidenza e supporto della conclusione trovata. Le regole trovate sono risultate tutte significative perché per ogni valore il *lift* è risultato essere maggiore di 1.

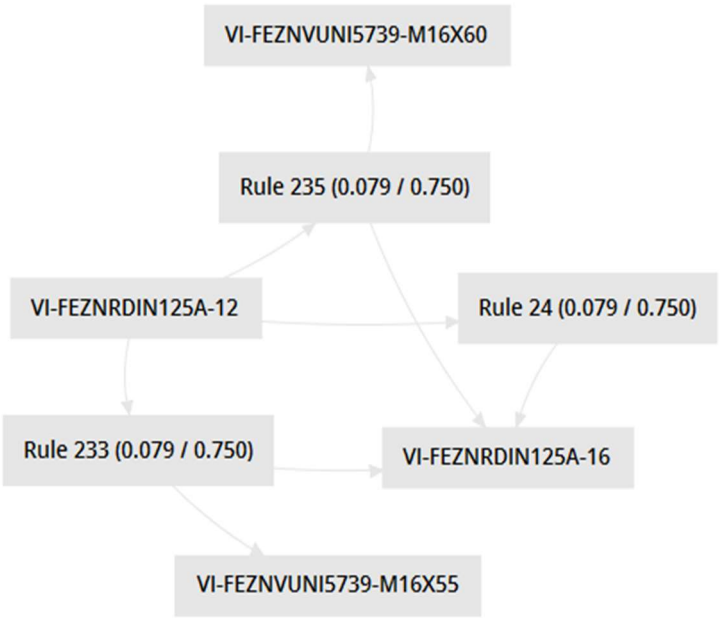

Figura 16: Esempio grafico di association rules

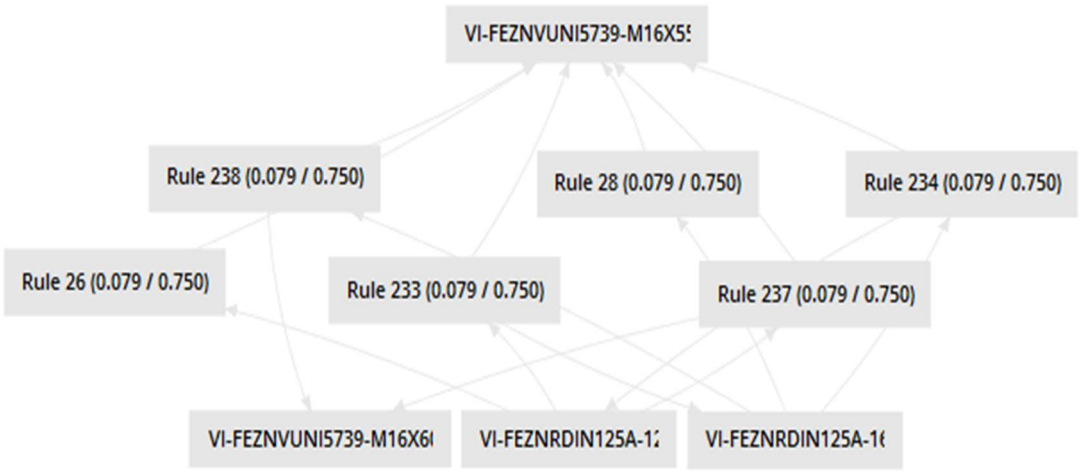

Figura 17: Esempio grafico di association rules

In tabella **m** ho inserito alcuni esempi di regole d'associazione trovati:

Premises	**Conclusion**	**Support**	**Confidence**	**Lift**
VI-FEZNRDIN125A-12	VI-FEZNVUNI5739-M16X60	0.079	0.75	9.49
VI-FEZNRDIN125A-12	VI-FEZNRDIN125A-16	0.079	0.75	9.49
VI-FEZNRDIN125A-12	VI-FEZNRDIN125A-16, VI-FEZNVUNI5739-M16X55	0.079	0.75	9.49
VI-FEZNRDIN125A-12	VI-FEZNVUNI5739-M16X55	0.079	0.75	9.49

Tabella m: Esempio di regole d'associazione

5.3 Association Rules Ordini di Conto Lavoro

Questa matrice è la più particolare. Il file sorgente comprende tutte le righe degli ordini Conto Lavoro. Questo tipo di ordini viene effettuato quando il cliente invia materiale di sua proprietà che deve essere lavorato dal fornitore. In questo caso il valore all'interno di ogni cella della matrice é *"true"* se la ragione sociale corrispondente ha effettuato almeno una volta una lavorazione sull'articolo conto terzi per chi utilizza *Concept*.

Anche in questo caso le dimensioni della matrice erano troppo grandi quindi ho cercato di "sfoltire" con logica come fatto per la matrice degli acquisti.

Itemset frequenti

Il supporto minimo trovato è 0.143. I dati sono risultati comunque molto elevati quindi ho dovuto effettuare un campionamento randomico del 50% in modo da rendere i tempi di calcolo più ragionevoli.
Il risultato ottenuto corrisponde a un totale di 188 *itemset* frequenti. Nella tabella **n** è riportata la distribuzione degli *itemset* in base alla dimensione, quindi al numero di articoli.

Size itemset	#itemset
1	13
2	28
3	53
4	55
5	31
6	7
7	1

Tabella n: Tabella itemset frequenti

Regole d'associazione

Per trovare le regole di associazione ho fissato come confidenza minima 0.7. Del campione preso in considerazione ci sono 13 articoli coinvolti che fanno parte di *itemset* frequenti e per i quali sono state trovate regole di associazione:

- 1000412044C01-AC
- 106372.0
- 106757.0
- 106758.0
- 106759.0
- 106760.0

- 106761.0
- 106762.0
- 106763.0
- 4.100064149E9
- 4100046399C01-AG
- 4100046399C01-AH
- 7.400024789E9

La maggior parte delle regole trovate sono significative perché il *lift* è maggiore di 1. Graficamente sono difficilmente interpretabili perché le regole trovate sono molte e le associazioni tra gli articoli sono quindi confusionarie (vedi figura **18**).

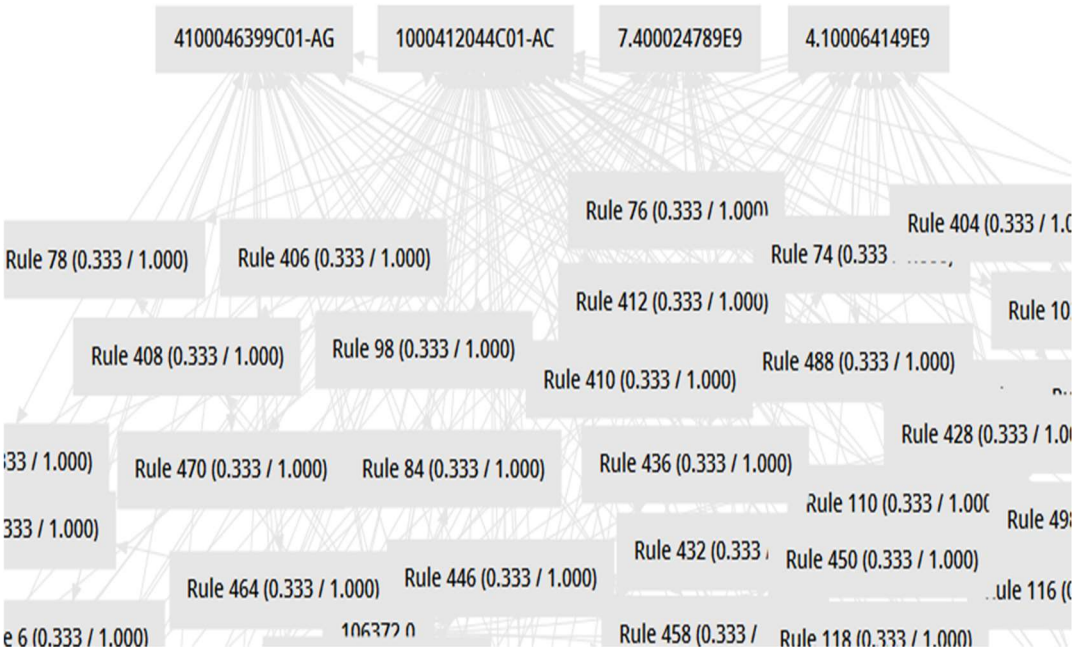

Figura 18: Esempio grafico di association rules

In tabella **o** ho inserito alcuni esempi delle regole di associazione trovate.

No.	Premises	Conclusion	Support	Confidence	Lift
93	1000412044C01-AC, 106372.0	7,40E+18	0.333	0.82	2.4
108	4.100064149E9, 4100046399C01-AH	1000412044C01-AC	0.333	0.82	2.4
1051	106759.0, 106762.0	106760.0, 106763.0	0.333	0.82	2.4
1099	4.100064149E9, 4100046399C01-AG, 7.400024789E9	4100046399C01-AH	0.333	0.82	2.4

Tabella o: Esempio di regole d'associazione

6. Classification

Classificazione e predizione sono processi che consistono nel creare dei modelli che possono essere usati per descrivere degli insiemi di dati e per fare previsioni future.

Classification e Clustering sono tecniche di analisi di dati che hanno differenze nella logica di base per come vengono applicate.
Nel caso di Classification si parla di *supervised learning*. I dati per trovare il modello sono stati preventivamente associati a etichette che indicano la classe di appartenenza, si tratta appunto di conoscenza supervisionata. I nuovi record infatti vengono segmentati assegnandogli etichette di *classe[8]*.

Per implementare questa tecnica di analisi ho applicato un processo a tre fasi:

1. Addestramento: si produce un modello da un *training set*. Ogni istanza in input fa parte di una tra un numero predefinito di classi. La classe per ogni attributo si trova consultando uno specifico attributo, il *class label attribute*. La classificazione è considerato un apprendimento "supervisionato" proprio perché l'etichetta che classifica il dato è conosciuta e viene fornita in input.

2. Stima dell'accuratezza: si stima l'accuratezza del modello usando un insieme di test, ad esempio, si misura la percentuale di errori commessi sull'insieme di test. È importante che l'insieme di test sia indipendente dall'insieme utilizzato per il campionamento.

3. Utilizzo del modello: si classificano istanze di classe ignota. Questo tipo di dati hanno la stessa struttura dei dati di training, l'unica differenza è che non si conosce l'attributo di classificazione e quindi si applica il modello per capire ogni dato a quale classe fa parte.

I modelli di classificazione basati su Decision Tree*[9]* sono considerati tra i migliori perché non sono costosi da costruire, sono facili da interpretare, sono facili da integrare con le basi di dati e hanno una buona accuratezza in molte applicazioni, anche in confronto ad altri metodi di classificazione.

L'algoritmo crea una struttura ad albero che somiglia ad un flow-chart:
- ogni *nodo interno* denota un test su un attributo e gli archi uscenti rappresentano i risultati del test.
- ogni *nodo foglia* rappresenta invece un'etichetta di classe o la distribuzione delle varie classi.

Il Decision Tree viene utilizzato sostanzialmente per due funzionalità specifiche:
- classificare un nuovo dato campione sulla base degli attributi, ovvero per assegnare un'etichetta di classe al nuovo dato
- effettuare test sui valori degli attributi del campione rispetto ai test presenti nel decision tree, quindi, a partire dalla radice, sulla base degli attributi del campione da classificare, segue un *cammino* fino ad una foglia. L'etichetta della foglia definisce la classe di appartenenza del campione.

Data l'elevata mole di dati, ho cercato di capire come gestirli in modo da poter ottenere risultati soddisfacenti. In prima analisi quindi, tramite interrogazioni SQL ho creato i file di training e di test per le tre tabelle che contenevano le righe degli ordini di vendita, acquisto e Conto Lavoro.
Per ognuna delle tabelle ho cercato quale fosse la ragione sociale più coinvolta nelle righe dei documenti. In questo modo ho trovato i clienti/fornitori con cui si hanno rapporti continui e ripetuti. Di seguito ho riportato nel dettaglio come ho strutturato le query e come ho creato i file per fare la classificazione tabella per tabella.
Dopodichè ho iniziato a processare effettivamente i dati in *Rapid Miner.* Inizialmente ho processato i dati di training e ho cercato il modello adatto attuando la fase di validazione. Una volta trovato il modello adatto l'ho applicato ai dati di test e ho visualizzato quindi i risultati ottenuti.

In figure **20**, **21, 22** e **23** ho riportato i processi fondamentali che ho utilizzato in questa fase:

- Cross Validation

Process

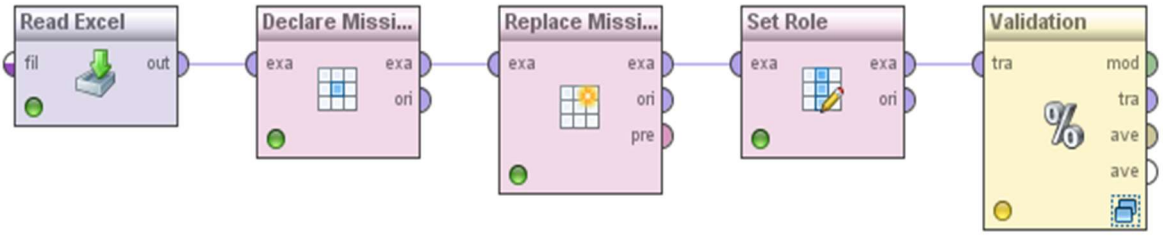

Figura 20: Processo Rapid Miner

Subprocess

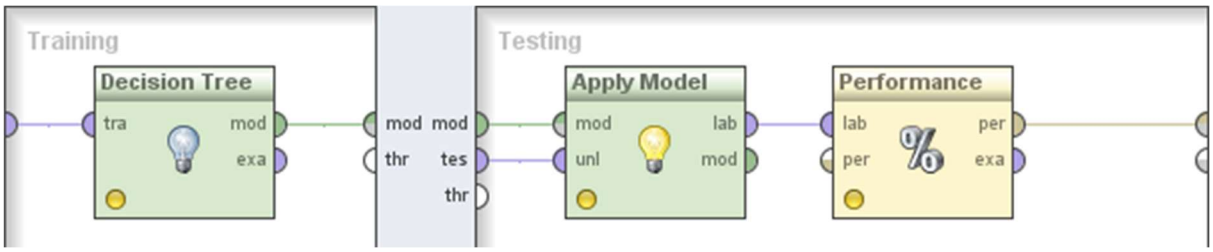

Figura 21: Sottoprocesso Rapid Miner

- Apply Model

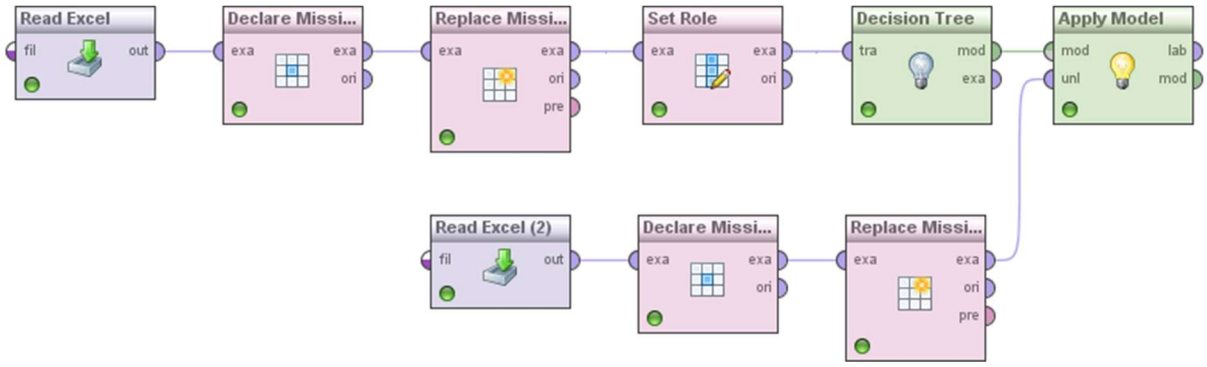

Figura 22: Applicazione modello Rapid Miner

- Correlation Matrix

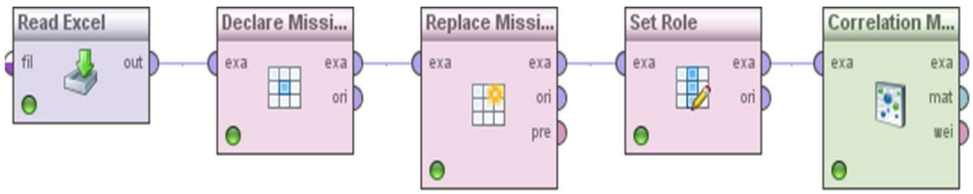

Figura 23: Correlation Matrix Rapid Miner

6.1 Classification Ordini di Vendita

Ho fatto le query a database per ottenere i dati degli articoli comprati e non comprati da una ragione sociale di cui non posso menzionare il nome

Articoli comprati:

```
SELECT DISTINCT ART.CDART, ART.ARDES, ART.ARUMS, ART.ARCLA,
ART.ARCOS, ART.ARRIC, ARPRZ, ART.ARDMX, ART.ARDMY, ART.ARDMZ,
ART.ARMAT, VER.DTRAS
FROM A01_ART_ICO ART, A01_SRC_DOC_VER VER
WHERE ART.CDART = VER.DRART
AND  (VER.DTANN = 2019 OR VER.DTANN = 2018)
AND VER.DRUMS = 'NR' AND VER.DTCAU LIKE '%ORD%' AND
VER.DTCAU<>'ORDIN'
AND VER.DTRAS='----------'
```

Articoli non comprati:

```
SELECT DISTINCT ART.CDART, ART.ARDES, ART.ARUMS, ART.ARCLA,
ART.ARCOS, ART.ARRIC, ARPRZ, ART.ARDMX, ART.ARDMY, ART.ARDMZ,
ART.ARMAT, VER.DTRAS
FROM A01_ART_ICO ART, A01_SRC_DOC_VER VER
WHERE ART.CDART = VER.DRART
AND  (VER.DTANN = 2019 OR VER.DTANN = 2018)
AND VER.DRUMS = 'NR' AND VER.DTCAU LIKE '%ORD%' AND
VER.DTCAU<>'ORDIN'
AND VER.DTRAS<>'----------'
```

Ho esportato i risultati in modo da poterli gestire in Excel. Ho aggiunto un attributo **Response** che nel primo caso ha valore '*Positive*' mentre nel secondo caso ha valore '*Negative*'.

Arrivato a questo punto ho creato i dati di *training*. Ho estratto casualmente 1000 record dalla tabella dei positivi e 1000 record dalla tabella dei negativi, li ho uniti in un'unica tabella, ho randomizzato le tuple ottenendo così il *Training Set*.

Dopodiché ho strutturato i dati di test. Per prima cosa ho preso a database tutti gli articoli che non erano presenti nei dataset precedenti attraverso la seguente query:

```
SELECT DISTINCT ART1.CDART, ART1.ARDES, ART1.ARUMS, ART1.ARCLA,
ART1.ARCOS, ART1.ARRIC, ARPRZ, ART1.ARDMX, ART1.ARDMY, ART1.ARDMZ,
ART1.ARMAT
FROM A01_ART_ICO ART1
WHERE ART1.CDART NOT IN (SELECT DISTINCT ART.CDART
                         FROM A01_ART_ICO ART, A01_SRC_DOC_VER VER
                         WHERE ART.CDART = VER.DRART
                         AND  (VER.DTANN = 2019 OR VER.DTANN = 2018)
                         AND VER.DRUMS = 'NR' AND VER.DTCAU LIKE '%ORD%'
AND
                         VER.DTCAU<>'ORDIN'
                         AND VER.DTRAS='--------------------------')
AND ART1.CDART NOT IN ( SELECT DISTINCT ART.CDART
                         FROM A01_ART_ICO ART, A01_SRC_DOC_VER VER
                         WHERE ART.CDART = VER.DRART
                         AND  (VER.DTANN = 2019 OR VER.DTANN = 2018)
                         AND VER.DRUMS = 'NR' AND VER.DTCAU LIKE '%ORD%'
AND
                         VER.DTCAU<>'ORDIN'
                         AND VER.DTRAS<>'--------------------------')
AND (ART1.ARCLA='PF' OR ART1.ARCLA='CL' OR ART1.ARCLA='SM')
AND ART1.ARMAT<>'NULL'
```

A questi dati ho aggiunto i dati che dalle tabelle precedenti non avevo estratto per ottenere il *Training Set*. La ragione per cui ho aggiunto questi dati è per ottenere una controprova sul modello generato: per questi dati l'etichetta di classificazione è conosciuta perché si conosce se si tratta di articoli che la controparte compra (Positive) oppure no (Negative) quindi una volta ottenuto i risultati ho verificato se effettivamente questi articoli venissero classificati coerentemente alle vendite reali. Il *Test Set* finale è composto da 29925 righe.

6.1.1 Cross Validation Ordini di Vendita

In questa fase ho cercato di validare il modello. Ho quindi modificato i parametri di profondità dell'albero e di *gain* minimo*[9]* (cioè l'attributo di split) e tramite la matrice di accuratezza ho scelto il modello più accurato.

1. Maximal depth 20, Minimal gain 0.03

accuracy: 77.63% +/- 2.07% (mikro: 77.63%)			
	true Positive	true Negative	class precision
pred. Positive	947	395	70.57%
pred. Negative	52	604	92.07%
class recall	94.79%	60.46%	

Figura 24: accuracy

Avg precision = 81.32% Avg recall = 77.62%. Non è un modello utilizzabile perché ha un valore di *precision* Positive e di *recall* Negative troppo basso.

2. Maximal depth 20, Minimal gain 0.02

accuracy: 81.48% +/- 2.98% (mikro: 81.48%)			
	true Positive	true Negative	class precision
pred. Positive	893	264	77.18%
pred. Negative	106	735	87.40%
class recall	89.39%	73.57%	

Figura 25: accuracy

Avg precision = 82.29% Avg recall = 81.48%

3. Maximal depth 20, Minimal gain 0.01

accuracy: 81.88% +/- 3.13% (mikro: 81.88%)			
	true Positive	true Negative	class precision
pred. Positive	910	273	76.92%
pred. Negative	89	726	89.08%
class recall	91.09%	72.67%	

Figura 26: accuracy

Avg precision = 83% Avg recall = 81.88%

4. Maximal depth 15, Minimal gain 0.03

accuracy: 77.63% +/- 2.07% (mikro: 77.63%)			
	true Positive	true Negative	class precision
pred. Positive	947	395	70.57%
pred. Negative	52	604	92.07%
class recall	94.79%	60.46%	

Figura 27: accuracy

Avg precision = 82.36% Avg recall = 81.13%. Non è un modello utilizzabile perché ha un valore di *precision* Positive e di *recall* Negative troppo basso.

5. Maximal depth 15, Minimal gain 0.02

accuracy: 81.13% +/- 3.19% (mikro: 81.13%)			
	true Positive	true Negative	class precision
pred. Positive	908	286	76.05%
pred. Negative	91	713	88.68%
class recall	90.89%	71.37%	

Figura 28: accuracy

Avg precision = 81.32% Avg recall = 77.62%

6. Maximal depth 15, Minimal gain 0.01

accuracy: 81.03% +/- 2.45% (mikro: 81.03%)			
	true Positive	true Negative	class precision
pred. Positive	917	297	75.54%
pred. Negative	82	702	89.54%
class recall	91.79%	70.27%	

Figura 29: accuracy

Avg precision = 82.54% Avg recall = 81.03%

Con decision tree modello più accurato con **Maximal depth: 20, Minimal gain: 0.01**

Ho inoltre effettuato lo stesso tipo di validazione con l'algoritmo K-NN per verificare quale fosse più accurato. Variando il parametro l'accuratezza aumenta fino al valore k=12 e poi diminuisce sempre. In ogni caso l'accuratezza non raggiunge mai i livelli di Decision Tree, per cui ho applicato ho applicato l'algoritmo ad albero.

L'attributo considerato più selettivo è **ARPRZ** cioè il prezzo dell'articolo perché applicando l'algoritmo di Decision Tree secondo le specifiche richieste questo attributo è il nodo radice dell'albero. Ciò significa che è l'attributo con il Gain Ratio più elevato.

6.1.2 Apply Model Ordini di Vendita

Applicando il modello ho quindi ottenuto il risultato di classificazione dei dati di test. Le righe totali erano 29925, di seguito ho riportato una tabella di esempio riportando un articolo per ogni intervallo di confidenze positive e negative trovate. Come previsto, il modello ha etichettato come certamente negativi (Confidence(Positive):0; Confidence(Negative):1) gli articoli che erano stati selezionati dalla query di articoli non venduti a quella controparte e come certamente positivi (Confidence(Positive):1; Confidence(Negative):0) gli articoli che erano stati selezionati dalla query di articoli venduti a quella controparte. tutti gli altri articoli hanno riportato valori intermedi che permettessero di fornire un'etichetta di classificazione con una certa probabilità. In questo modo gli articoli etichettati in positivo sono prodotti che eventualmente possono essere presi in considerazione da inserire in un'offerta commerciale per il cliente su cui è stata fatta l'analisi.

+	-	Prediction	CDART	ARDES	ARMAT
0.00	1.00	Negative	**8138167A**	PARTICOLARE A DISEGNO - 1,50 MM	AISI430SATP
0.01	0.99	Negative	**DOCA-03365**	PONTE SQUAMATORE CR93	DECAPATO
0.12	0.88	Negative	**MEXX0192FSXX 00-02**	PARETE SUPERIORE L0686-M003	AL5754H111
0.12	0.88	Negative	**2562-A2200-A0-00-P2**	PARTICOLARE A DISEGNO - 1,50 MM	DECAPATO
0.15	0.85	Negative	**AEM-KM-02107-V001**	GRUNDPLATTE	-

0.17	0.83	Negative	**2M3731**	PIASTRINA DI CHIUSURA	DECAPATO
0.25	0.75	Negative	**1561-A595E-00-0-P1**	PARTICOLARE A DISEGNO - 1,50 MM	ZINCATO
0.33	0.66	Negative	**BC016747**	NASTRO 2248X620 1X220V 50HZ 85W	-
0.62	0.38	Positive	**TARGHETTE63-02**	PARTICOLARE A DISEGNO - 1,00 MM	AISI304SATP
0.66	0.34	Positive	**PS02560G**	GIUNZIONE X TRAVETTA DI 120	NERO
0.77	0.23	Positive	**8736256.0**	PIAST SP. 2.5 ZINC L181 XGUID PIAN	ZINCATO
0.85	0.15	Positive	**FO-707**	PROTOTIPO NALDI 650	-
0.85	0.15	Positive	**P25-42529F**	FD PL H200/275 L1330 /B75 LF	LUCIDO
1.00	0.00	Positive	**670M30001B-1**	RIVESTIMENTO PIANO	AISI304BAP

Tabella p: Tabella d'esempio classificazioni

6.1.3 Correlation Matrix Ordini di Vendita

Dalla matrice di correlazione*[10]* degli attributi si nota che ci sono molti attributi correlati. In particolare ci sono due coppie altamente correlate:

- CDART-ARDES (coeff 0.956): codice e descrizione articolo
- ARCOS-ARRIC (coeff 0.771): costo e prezzo dell'articolo

Due coppie mediamente correlate:

- ARDMX-ARDMY (coeff 0.434): larghezza e lunghezza articolo
- ARDMZ-ARMAT (coeff 0.471): spessore e materiale articolo

Attributes	CDART	ARDES	ARUMS	ARCLA	ARCOS	ARRIC	ARPRZ	ARDMX	ARDMY	ARDMZ	ARMAT
CDART	1	0.956	?	0.027	0.006	-0.016	0.016	0.000	-0.005	-0.001	-0.010
ARDES	0.956	1	?	0.030	0.011	-0.005	0.020	-0.007	-0.022	-0.002	-0.024
ARUMS	?	?	?	?	?	?	?	?	?	?	?
ARCLA	0.027	0.030	?	1	-0.010	-0.022	-0.007	-0.059	-0.061	-0.044	-0.051
ARCOS	0.006	0.011	?	-0.010	1	-0.031	0.771	-0.076	-0.069	-0.049	-0.073
ARRIC	-0.016	-0.005	?	-0.022	-0.031	1	-0.005	-0.041	-0.026	-0.018	0.021
ARPRZ	0.016	0.020	?	-0.007	0.771	-0.005	1	-0.056	-0.049	-0.033	-0.054
ARDMX	0.000	-0.007	?	-0.059	-0.076	-0.041	-0.056	1	0.434	0.151	0.243
ARDMY	-0.005	-0.022	?	-0.061	-0.069	-0.026	-0.049	0.434	1	0.137	0.272
ARDMZ	-0.001	-0.002	?	-0.044	-0.049	-0.018	-0.033	0.151	0.137	1	0.471
ARMAT	-0.010	-0.024	?	-0.051	-0.073	0.021	-0.054	0.243	0.272	0.471	1

Figura 30: correlation matrix

Naive Bayes è un algoritmo di *supervised learning* adatto a risolvere problemi di classificazione binari (a due classi) e multi-classe. La peculiarità principale dell'algoritmo, oltre ad avvalersi del teorema di Bayes (probabilità condizionata), è che si basa sul fatto che tutte le caratteristiche non siano correlate l'una all'altra. La presenza o l'assenza di una caratteristica non influenzano la presenza o l'assenza di altre. Come si può notare questo algoritmo non può essere utilizzato perché non è verificata l'ipotesi d'indipendenza tra gli attributi.

6.2 Classification Ordini di Conto Lavoro

ho fatto le query a database per ottenere i dati degli articoli comprati e non comprati dalla ragione sociale di cui non posso menzionare il nome.

Articoli inviati in Conto Lavoro a quel fornitore:

```
SELECT DISTINCT ICO.CDART, ICO.ARDES, ICO.ARUMS, ICO.ARCLA,
ICO.ARCOS, ICO.ARRIC, ICO.ARPRZ, ICO.ARDMX, ICO.ARDMY, ICO.ARDMZ,
ICO.ARMAT
FROM A01_SRC_DOC_CTR CTR, A01_ART_ICO ICO
WHERE  CTR.DRART = ICO.CDART
AND (DTANN='2018' OR DTANN='2019')
AND DTCAU LIKE '%ORD%'
AND CTR.DTRAS = '----------------------------------------'
```

Articoli inviati in Conto Lavoro ad altri fornitori:

```
SELECT DISTINCT ICO.CDART, ICO.ARDES, ICO.ARUMS, ICO.ARCLA,
ICO.ARCOS, ICO.ARRIC, ICO.ARPRZ, ICO.ARDMX, ICO.ARDMY, ICO.ARDMZ,
ICO.ARMAT
FROM A01_SRC_DOC_CTR CTR, A01_ART_ICO ICO
WHERE  CTR.DRART = ICO.CDART
```

```
AND (DTANN='2018' OR DTANN='2019')
AND DTCAU LIKE '%ORD%'
AND CTR.DTRAS <> '----------'
```

Ho esportato i risultati in modo da poterli gestire in Excel. Ho aggiunto un attributo **Response** che nel primo caso ha valore '*Positive*' mentre nel secondo caso ha valore '*Negative*'.

Arrivato a questo punto ho creato i dati di *training*. Ho estratto casualmente 500 record dalla tabella dei positivi e 500 record dalla tabella dei negativi, li ho uniti in un'unica tabella, ho randomizzato le tuple ottenendo così il *Training Set*.

Dopodiché ho strutturato i dati di test. Per prima cosa ho preso a database tutti gli articoli che non erano presenti nei dataset precedenti attraverso la seguente query:

```
SELECT DISTINCT ICO1.CDART, ICO1.ARDES, ICO1.ARUMS, ICO1.ARCLA,
ICO1.ARCOS, ICO1.ARRIC,
ICO1.ARPRZ,ICO1.ARDMX,ICO1.ARDMY,ICO1.ARDMZ,ICO1.ARMAT
FROM A01_SRC_DOC_CTR CTR1, A01_ART_ICO ICO1
WHERE CTR1.DRART NOT IN (SELECT DISTINCT CTR.DRART
                         FROM A01_SRC_DOC_CTR CTR, A01_ART_ICO ICO
                         WHERE  CTR.DRART = ICO.CDART
                         AND (DTANN='2018' OR DTANN='2019')
                         AND DTCAU LIKE '%ORD%'
                         AND ICO.ARMAT <> 'NULL'
                         AND CTR.DTRAS<> '----------')
AND CTR1.DRART NOT IN (SELECT DISTINCT CTR.DRART
                         FROM A01_SRC_DOC_CTR CTR, A01_ART_ICO ICO
                         WHERE  CTR.DRART = ICO.CDART
                         AND (DTANN='2018' OR DTANN='2019')
                         AND DTCAU LIKE '%ORD%'
                         AND ICO.ARMAT <> 'NULL'
                         AND CTR.DTRAS= '----------')
AND ICO1.ARMAT <> 'NULL' AND ICO1.ARMAT <> ''
```

A questi dati ho aggiunto i dati che dalle tabelle precedenti non avevo estratto per ottenere il *Training Set*. La ragione per cui ho aggiunto questi dati è per ottenere una controprova sul modello generato: per questi dati l'etichetta di classificazione è conosciuta perché si conosce se si tratta di articoli che la controparte compra (Positive) oppure no (Negative) quindi una volta ottenuto i risultati ho verificato se effettivamente questi articoli

venissero classificati coerentemente alle vendite reali. Il *Test Set* finale è composto da 35988 righe.

6.2.1 Cross Validation Ordini di Conto Lavoro

In questa fase ho cercato di validare il modello. Ho quindi modificato i parametri di profondità dell'albero e di *gain* minimo[9] (cioè l'attributo di split) e tramite la matrice di accuratezza ho scelto il modello più accurato da applicare.

1. Maximal depth 20, Minimal gain 0.03

accuracy: 63.16% +/- 4.52% (mikro: 63.17%)

	true Negative	true Positive	class precision
pred. Negative	217	86	71.62%
pred. Positive	283	416	59.51%
class recall	43.40%	82.87%	

Figura 31: accuracy

Avg precision = 65.56% Avg recall = 63.13%. Non è un modello utilizzabile perché ha un valore *precision* Positive troppo basso.

2. Maximal depth 20, Minimal gain 0.02

accuracy: 63.37% +/- 4.51% (mikro: 63.37%)

	true Negative	true Positive	class precision
pred. Negative	184	51	78.30%
pred. Positive	316	451	58.80%
class recall	36.80%	89.84%	

Figura 32 accuracy

Avg precision = 68.4% Avg recall = 63.32%. Non è un modello utilizzabile perché ha un valore *recall* Negative e di *precision* Positive troppo basso.

3. Maximal depth 20, Minimal gain 0.01

accuracy: 65.96% +/- 4.81% (mikro: 65.97%)			
	true Negative	true Positive	class precision
pred. Negative	217	58	78.91%
pred. Positive	283	444	61.07%
class recall	43.40%	88.45%	

Figura 33: accuracy

Avg precision = 69.99% Avg recall = 65.92%.

4. Maximal depth 15, Minimal gain 0.03

accuracy: 63.16% +/- 4.52% (mikro: 63.17%)			
	true Negative	true Positive	class precision
pred. Negative	217	86	71.62%
pred. Positive	283	416	59.51%
class recall	43.40%	82.87%	

Figura 34: accuracy

Avg precision = 65.56% Avg recall = 63.13%

5. Maximal depth 15, Minimal gain 0.02

accuracy: 63.37% +/- 4.51% (mikro: 63.37%)			
	true Negative	true Positive	class precision
pred. Negative	184	51	78.30%
pred. Positive	316	451	58.80%
class recall	36.80%	89.84%	

Figura 35: accuracy

Avg precision = 65.56% Avg recall = 63.13%. Non è un modello utilizzabile perché ha un valore *recall* Negative e di *precision* Positive troppo basso.

6. Maximal depth 15, Minimal gain 0.01

accuracy: 65.76% +/- 4.79% (mikro: 65.77%)

	true Negative	true Positive	class precision
pred. Negative	217	60	78.34%
pred. Positive	283	442	60.97%
class recall	43.40%	88.05%	

Figura 36: accuracy

Avg precision = 69.65% Avg recall = 65.72%

Con decision tree[9] modello più accurato con **Maximal depth 20, Minimal gain 0.01**

Ho inoltre effettuato lo stesso tipo di validazione con l'algoritmo K-NN per verificare quale fosse più accurato. Variando il parametro l'accuratezza aumenta fino al valore k=7 e poi diminuisce sempre. In ogni caso l'accuratezza non raggiunge mai i livelli di Decision Tree, per cui ho applicato ho applicato l'algoritmo ad albero.

L'attributo considerato più selettivo è **ARCOS** cioè il costo standard dell'articolo perché applicando l'algoritmo di Decision Tree secondo le specifiche richieste questo attributo è il nodo radice dell'albero. Ciò significa che è l'attributo con il Gain Ratio più elevato.

6.2.2 Apply Model Ordini di Conto Lavoro

Applicando il modello ho quindi ottenuto il risultato di classificazione dei dati di test. Le righe totali erano 35988, di seguito ho riportato una tabella di esempio riportando un articolo per ogni intervallo di confidenze positive e negative trovate. Come previsto, il modello ha etichettato come certamente negativi (Confidence(Positive):0; Confidence(Negative):1) gli articoli che erano stati selezionati dalla query di articoli non venduti a quella controparte e come certamente positivi (Confidence(Positive):1; Confidence(Negative):0) gli articoli che erano stati selezionati dalla query di articoli venduti a quella controparte. tutti gli altri articoli hanno riportato

valori intermedi che permettessero di fornire un'etichetta di classificazione con una certa probabilità. In questo modo gli articoli etichettati in positivo sono prodotti che eventualmente possono essere presi in considerazione da inserire in un'offerta commerciale per il cliente su cui è stata fatta l'analisi.

+	-	Prediction	CDART	ARDES	ARMAT
0.0	1.0	Negative	**4200052724C 01-AC**	PIATTO-ENDLAGERPLATTE-EL	NERO
0.045	0.95	Negative	**7373-PR12**	PARTICOLARE A DISEGNO - 5,00 MM	DECAPATO S235JR
0.3	0.7	Negative	**BC005724A**	BC005724A	AISI430SAT P
0.089	0.91	Negative	**EO-002**	LAMIERA C70 BONIFICATO RULLI INFERIORE	C70B
0.12	0.88	Negative	**OCN1352-P3**	PARTICOLARE A DISEGNO - AISI3042B - 2,00 MM	AISI3042B
0.25	0.75	Negative	**PL-0011C-P1**	PARTICOLARE A DISEGNO - DECAPATOS235JR - 5,00 MM	DECAPATO S235JR
0.75	0.25	Positive	**106100.0**	LAMIERA DI BASE	S355J2
1.0	0.0	Positive	**01S287-7-P31**	PARTICOLARE A DISEGNO - 8,00 MM	S355J2

Tabella q: Tabella d'esempio classificazioni

6.2.3 Correlation Matrix Ordini di Conto Lavoro

Dalla matrice di correlazion degli attributi si nota che ci sono molti attributi correlati. In particolare ci sono due coppie altamente correlate:

- CDART-ARDES (coeff 0.973): codice e descrizione articolo
- ARCOS-ARRIC (coeff 0.950): costo e prezzo dell'articolo

Due coppie mediamente correlate:

- ARDMX-ARDMY (coeff 0.525): larghezza e lunghezza articolo

Attributes	CDART	ARDES	ARUMS	ARCLA	ARCOS	ARRIC	ARPRZ	ARDMX	ARDMY	ARDMZ	ARMAT
CDART	1	0.973	-0.051	0.017	-0.025	0.023	-0.025	-0.010	-0.033	-0.035	0.017
ARDES	0.973	1	-0.051	0.026	-0.023	0.027	-0.023	-0.013	-0.041	-0.043	0.010
ARUMS	-0.051	-0.051	1	0.100	-0.001	-0.003	-0.001	-0.014	-0.013	-0.016	-0.021
ARCLA	0.017	0.026	0.100	1	-0.034	0.049	-0.032	-0.063	-0.058	-0.024	-0.004
ARCOS	-0.025	-0.023	-0.001	-0.034	1	-0.005	0.990	-0.016	-0.015	-0.019	-0.026
ARRIC	0.023	0.027	-0.003	0.049	-0.005	1	-0.004	-0.022	-0.026	-0.026	-0.030
ARPRZ	-0.025	-0.023	-0.001	-0.032	0.990	-0.004	1	-0.010	-0.014	-0.017	-0.023
ARDMX	-0.010	-0.013	-0.014	-0.063	-0.016	-0.022	-0.010	1	0.525	0.284	0.327
ARDMY	-0.033	-0.041	-0.013	-0.058	-0.015	-0.026	-0.014	0.525	1	0.290	0.283
ARDMZ	-0.035	-0.043	-0.016	-0.024	-0.019	-0.026	-0.017	0.284	0.290	1	0.625
ARMAT	0.017	0.010	-0.021	-0.004	-0.026	-0.030	-0.023	0.327	0.283	0.625	1

Figura 37: correlation matrix

Come detto in precedenza l'algoritmo di Naive Bayes non può essere applicato a causa della dipendenza tra attributi.

6.3 Classification Ordini di Acquisto

Con questa tabella i dati ottenuti sono stati molto inferiori rispetto ai precedenti quindi il modello è più una prototipazione che può essere utilizzata come base di partenza in caso di tabella con molti più dati ma che necessiterebbe di una revisione nell'applicazione dell'algoritmo. Per fare le query ho considerato una ragione sociale di cui non posso menzionare il nome.

Articoli acquistati:

```
SELECT DISTINCT ICO1.CDART, ICO1.ARDES, ICO1.ARUMS, ICO1.ARCLA,
ICO1.ARCOS, ICO1.ARRIC,
ICO1.ARPRZ,ICO1.ARDMX,ICO1.ARDMY,ICO1.ARDMZ,ICO1.ARMAT
FROM A01_ART_ICO ICO1, A01_SRC_DOC_ACR ACR
WHERE ICO1.CDART = ACR.DRART
AND (DTANN=2019 OR DTANN=2018)
AND DTCAU LIKE '%ORD%'
AND DTRAS='----------'
```

Articoli non acquistati:

```
SELECT DISTINCT ICO1.CDART, ICO1.ARDES, ICO1.ARUMS, ICO1.ARCLA,
ICO1.ARCOS, ICO1.ARRIC,
ICO1.ARPRZ,ICO1.ARDMX,ICO1.ARDMY,ICO1.ARDMZ,ICO1.ARMAT
FROM A01_ART_ICO ICO1, A01_SRC_DOC_ACR ACR
WHERE ICO1.CDART = ACR.DRART
AND (DTANN=2019 OR DTANN=2018)
AND DTCAU LIKE '%ORD%'
AND DTRAS<>'----------'
```

Ho esportato i risultati in modo da poterli gestire in Excel. Ho aggiunto un attributo **Response** che nel primo caso ha valore '*Positive*' mentre nel secondo caso ha valore '*Negative*'.

Arrivato a questo punto ho creato i dati di *training*. Ho estratto casualmente 100 record dalla tabella dei positivi e 100 record dalla tabella dei negativi, li ho uniti in un'unica tabella, ho randomizzato le tuple ottenendo così il *Training Set*.

Dopodiché ho strutturato i dati di test. Per prima cosa ho preso a database tutti gli articoli che non erano presenti nei dataset precedenti attraverso la seguente query:

```
SELECT DISTINCT ICO1.CDART, ICO1.ARDES, ICO1.ARUMS, ICO1.ARCLA,
ICO1.ARCOS, ICO1.ARRIC,
ICO1.ARPRZ,ICO1.ARDMX,ICO1.ARDMY,ICO1.ARDMZ,ICO1.ARMAT
FROM A01_ART_ICO ICO1, A01_SRC_DOC_ACR ACR
WHERE ICO1.CDART = ACR.DRART
AND (DTANN=2019 OR DTANN=2018)
AND DTCAU LIKE '%ORD%'
AND ICO1.CDART NOT IN (SELECT DISTINCT ICO.CDART
                       FROM A01_ART_ICO ICO, A01_SRC_DOC_ACR ACR
                       WHERE ICO.CDART = ACR.DRART
                       AND (DTANN=2019 OR DTANN=2018)
                       AND DTCAU LIKE '%ORD%'
                       AND DTRAS='----------')
AND ICO1.CDART NOT IN (SELECT DISTINCT ICO.CDART
                       FROM A01_ART_ICO ICO, A01_SRC_DOC_ACR ACR
                       WHERE ICO.CDART = ACR.DRART
                       AND (DTANN=2019 OR DTANN=2018)
                       AND DTCAU LIKE '%ORD%'
                       AND DTRAS<>'----------')
```

A questi dati ho aggiunto i dati che dalle tabelle precedenti non avevo estratto per ottenere il *Training Set*. La ragione per cui ho aggiunto questi dati è per ottenere una controprova sul modello generato: per questi dati l'etichetta di classificazione è conosciuta perché si conosce se si tratta di articoli che la controparte compra (Positive) oppure no (Negative) quindi una volta ottenuto i risultati ho verificato se effettivamente questi articoli venissero classificati coerentemente alle vendite reali. Il *Test Set* finale è composto da 2808 righe.

6.3.1 Cross Validation Ordini di Acquisto

1. Maximal depth 20, Minimal gain 0.03

accuracy: 70.21% +/- 6.45% (mikro: 70.20%)			
	true Positive	true Negative	class precision
pred. Positive	95	55	63.33%
pred. Negative	4	44	91.67%
class recall	95.96%	44.44%	

Figura 38: accuracy

Avg precision = 77.5% Avg recall = 70.2%. Non è un modello utilizzabile perché ha un valore *recall* Negative.

2. Maximal depth 20, Minimal gain 0.02

accuracy: 71.21% +/- 5.49% (mikro: 71.21%)			
	true Positive	true Negative	class precision
pred. Positive	95	53	64.19%
pred. Negative	4	46	92.00%
class recall	95.96%	46.46%	

Figura 39: accuracy

Avg precision = 78.1% Avg recall = 71.21%

3. Maximal depth 20, Minimal gain 0.01

accuracy: 71.21% +/- 5.49% (mikro: 71.21%)			
	true Positive	true Negative	class precision
pred. Positive	95	53	64.19%
pred. Negative	4	46	92.00%
class recall	95.96%	46.46%	

Figura 40: accuracy

Avg precision = 78.1% Avg recall = 71.21%

4. Maximal depth 15, Minimal gain 0.03

accuracy: 70.21% +/- 6.45% (mikro: 70.20%)			
	true Positive	true Negative	class precision
pred. Positive	95	55	63.33%
pred. Negative	4	44	91.67%
class recall	95.96%	44.44%	

Figura 41: accuracy

Avg precision = 77.5% Avg recall = 70.2%. Non è un modello utilizzabile perché ha un valore *recall* Negative.

5. Maximal depth 15, Minimal gain 0.02

accuracy: 71.21% +/- 5.49% (mikro: 71.21%)			
	true Positive	true Negative	class precision
pred. Positive	95	53	64.19%
pred. Negative	4	46	92.00%
class recall	95.96%	46.46%	

Figura 42: accuracy

Avg precision = 78.1% Avg recall = 71.21%

6. Maximal depth 15, Minimal gain 0.01

accuracy: 71.21% +/- 5.49% (mikro: 71.21%)			
	true Positive	true Negative	class precision
pred. Positive	95	53	64.19%
pred. Negative	4	46	92.00%
class recall	95.96%	46.46%	

Figura 43: accuracy

Avg precision = 78.1% Avg recall = 71.21%

Utilizzando invece l'algoritmo kNN la matrice di accuratezza ha dato risultati più positivi rispetto ai tentativi precedenti. Nei problemi di classificazione, il KNN determina l'etichetta di classe prevista, valutando un tipo di distanza.

Quando *k è piccolo*, stiamo limitando la regione di una determinata previsione e costringendo il nostro classificatore ad essere "più cieco" rispetto alla distribuzione generale. Al contrario, un *k grande* riduce l'impatto della varianza causato da un errore casuale, ma corre il rischio di ignorare piccoli dettagli che potrebbero essere rilevanti.

1. K=3

accuracy: 81.29% +/- 5.10% (mikro: 81.31%)

	true Positive	true Negative	class precision
pred. Positive	85	23	78.70%
pred. Negative	14	76	84.44%
class recall	85.86%	76.77%	

Figura 44: accuracy

Avg precision = 81.57% Avg recall = 81.31%

A parità di accuracy scelgo quello con minimal gain e maximal depth più restrittivi, quindi il modello selezionato è **Maximal depth 15, Minimal gain 0.02.**

6.3.2 Apply Model Ordini di Acquisto

+	-	Prediction	CDART	ARDES
0.0	1.0	Negative	**VI-IXA4DUNI5588-M12**	DADO ESAGONALE MEDIO UNI 5588 AISI 316 M12
0.0	1.0	Negative	**AI-MASCHERA-V**	MASCHERINE PIEGHEVOLI CON VALVOLA
0.07	0.93	Negative	**MAG-PL-RULL**	PALLET PER MAGAZZINO 505X60X21 PER RULLIERE
0.07	0.93	Negative	**4100137618C01-AA-EST**	TAGLIO KIT COMPOSTO DAI DUE PEZZI NEL DXF S235 SP. 30
0.65	0.35	Positive	**490-04127**	CATALIZZATORE EPOX A.S
0.65	0.35	Positive	**TR-FES235DEC-25X15X1,5**	TUBOLARE RETTANGOLARE FERRO DECAPATO S235 25X15X1,5

Tabella r: Tabella d'esempio classificazioni

L'attributo considerato più selettivo è **ARPRZ** cioè il prezzo dell'articolo perché applicando l'algoritmo di Decision Tree secondo le specifiche richieste questo attributo è il nodo radice dell'albero. Ciò significa che è l'attributo con il Gain Ratio più elevato.

6.3.3 Correlation Matrix Ordini di Acquisto

Dalla matrice di correlazione degli attributi si nota che ci sono molti attributi correlati. In particolare ci sono due coppie altamente correlate:

- CDART-ARDES (coeff 1.000): codice e descrizione articolo
- ARUMS-ARCLA (coeff 0.780): unità di misura e classe dell'articolo
- ARPRZ-ARCOS (coeff 1.000): costo e prezzo dell'articolo

Due coppie mediamente correlate:

- ARDMZ-ARMAT (coeff 0.517): spessore e materiale dell'articolo

Attributes	CDART	ARDES	ARUMS	ARCLA	ARCOS	ARRIC	ARPRZ	ARDMX	ARDMY	ARDMZ	ARMAT
CDART	1	1.000	0.147	0.162	-0.038	0.073	-0.036	?	?	0.116	0.183
ARDES	1.000	1	0.146	0.162	-0.038	0.073	-0.036	?	?	0.116	0.182
ARUMS	0.147	0.146	1	0.780	0.136	-0.162	0.134	?	?	0.402	0.490
ARCLA	0.162	0.162	0.780	1	0.240	-0.163	0.238	?	?	0.196	0.225
ARCOS	-0.038	-0.038	0.136	0.240	1	0.046	1.000	?	?	-0.035	-0.045
ARRIC	0.073	0.073	-0.162	-0.163	0.046	1	0.056	?	?	0.063	0.022
ARPRZ	-0.036	-0.036	0.134	0.238	1.000	0.056	1	?	?	-0.034	-0.044
ARDMX	?	?	?	?	?	?	?	?	?	?	?
ARDMY	?	?	?	?	?	?	?	?	?	?	?
ARDMZ	0.116	0.116	0.402	0.196	-0.035	0.063	-0.034	?	?	1	0.517
ARMAT	0.183	0.182	0.490	0.225	-0.045	0.022	-0.044	?	?	0.517	1

Figura 45: correlation matrix

Conclusioni

Arrivati alla conclusione di questo progetto bisogna capire quali sono le positività e le negatività delle risultanze trovate, per trovare eventualmente quali possano essere le migliorie o le potenzialità che potrebbe avere un lavoro più specifico e strutturato.

In questo progetto di tesi sono state implementate le tecniche principali di *data mining* ma non ha la presunzione di racchiudere tutte le informazioni utili estrapolabili da un DB di questa portata. Le tabelle che hanno un carattere rilevante all'interno di questo DB sono centinaia e quindi il lavoro che è stato fatto è infinitesimale rispetto al potenziale disponibile.

Le operazioni di *Data Mining* hanno una finalità ben precisa, quella di ottenere risultati che portino a scelte di business utili per l'impresa che ha il possesso dei dati analizzati.

Prendendo ad esempio la parte di *Classification* è evidente come, correttamente configurata, è una tecnica che può dare informazioni estremamente utili per il cliente perché permette di trovare una conoscenza intrinseca ai dati stessi che il cliente ha in possesso e che non sa di avere in mancanza di un modello di classificazione adeguato. Nell'analisi effettuata infatti è molto interessante il fatto di aver trovato un'etichetta positiva o negativa in base a una precisa ragione sociale. Sostanzialmente grazie a questo tipo di modello ho trovato se un articolo è vendibile a quel specifico cliente o acquistabile da quel specifico fornitore con una certa probabilità. Questo tipo di informazione può essere estremamente utile a livello commerciale.

Questo tipo di progetto rappresenta quindi un prototipo su cui lavorare. Applicando queste tecniche in maniera mirata, personalizzando cliente per cliente in base alle esigenze, sarebbe possibile ottenere un servizio vendibile come modulo aggiuntivo al software ERP. Questo tipo di

servizio rappresenta sicuramente un valore aggiunto per il cliente il quale ha la possibilità di avere una conoscenza direttamente dai propri dati.

Note

[1] Introduction to Data Mining, Vipin Kumar – p. 125

[2] Introduction to Data Mining, Vipin Kumar – p. 131

[3] Introduction to Data Mining, Vipin Kumar – p. 127

[4] Introduction to Data Mining, Vipin Kumar – p. 147

[5] Introduction to Data Mining, Vipin Kumar – p. 76

[6] Introduction to Data Mining, Vipin Kumar – p. 79

[7] Introduction to Data Mining, Vipin Kumar – p. 77

[8] Introduction to Data Mining, Vipin Kumar – p. 29

[9] Introduction to Data Mining, Vipin Kumar – p. 39

[10] Introduction to Data Mining, Vipin Kumar – p. 89

Sitografia

- *http://dbdmg.polito.it/wordpress/teaching/business-intelligence/*

- *http://dbdmg.polito.it/wordpress/teaching/data-science-e-tecnologie-2019/*

- *http://dbdmg.polito.it/wordpress/teaching/data-science-and-database-technology/*

Bibliografia

- Rapid Miner Studio Manual

- Introduction to Machine Learning with Python, Andreas C. Müller & Sarah Guido

- Introduction to Data Mining, Vipin Kumar

Tools

- Microsoft SQL Server 2016

- Microsoft SQL Server Management Studio 2017

- Microsoft Excel

- Microsoft Word

- Rapid Miner 5

- Rapid Miner 9