



UNIVERSITÀ DI PISA

DIPARTIMENTO DI INGEGNERIA DELL'ENERGIA DEI SISTEMI
DEL TERRITORIO E DELLE COSTRUZIONI

RELAZIONE PER IL CONSEGUIMENTO DELLA
LAUREA MAGISTRALE IN INGEGNERIA GESTIONALE

***Design, Development and Validation
of a Decision Support System for optimal spare part
management in maintenance activities***

SINTESI

RELATORI

Prof. Ing. Francesco Marcelloni
Dipartimento di Ingegneria Dell'Informazione

Dott. Fabrizio Ruffini
Dipartimento di Ingegneria Dell'Informazione

Dott. Jérémy Decerle
LogObject AG

IL CANDIDATO

Giovanna Tascini
g.tascini@studenti.unipi.it

Design, Development and Validation of a Decision Support System for optimal spare part management in maintenance activities

Giovanna Tascini

Sommario

Questa tesi è stata realizzata in collaborazione con il *team* di R&S dell'azienda LogObject AG ed eseguita in parte nella sede di Zurigo, nell'ambito del bando di tesi all'estero. L'obiettivo principale del progetto è lo sviluppo di un sistema in grado di prevedere la tipologia di materiali e pezzi di ricambio necessari alla manutenzione di elettrodomestici, prodotti da un cliente di LogObject. I dati, analizzati mediante linguaggio Python, sono stati estratti dal software *mLogistics*, sviluppato da LogObject e impiegato dal cliente per la gestione delle attività degli operatori responsabili della manutenzione. Durante il progetto ho svolto le seguenti attività: acquisizione ed esplorazione dei dati estratti dal software gestionale; sviluppo di tre modelli predittivi principali basati su *text mining*, *machine learning* e *frequent patterns mining*; confronto dei diversi modelli e identificazione del più performante. Le analisi hanno condotto alla creazione di un sistema versatile e con ampie possibilità di applicazione anche per altri clienti dell'azienda, in grado di aumentare l'efficienza nelle attività di manutenzione. Sono state inoltre poste le basi per possibili futuri miglioramenti del progetto, a partire dalla proposta di una nuova modalità di raccolta dei dati.

Abstract

This thesis was performed in collaboration with the R&D team of LogObject AG and partly carried out in the Zurich office. The main goal of the project is the development of a system able to predict the specific type of materials and spare parts necessary for the maintenance of household appliances, produced by a LogObject customer. The data, analysed with Python, was extracted from the *mLogistics* software, developed by LogObject and used by the customer to manage the activities of the operators in maintenance tasks. During the project I achieved the following activities: acquisition and exploration of data extracted from the software; development of three main predictive models based on *text mining*, *machine learning* and *frequent patterns mining*; comparison of different models and identification of the best performing one. The analyses led to the creation of a versatile system with wide application possibilities for the other customers of the company, able to increase the efficiency in maintenance activities. In addition, the basis for possible future improvements of the project has been laid, starting with the proposal for a new data collection method.

1. Introduzione

LogObject AG è fornitrice di servizi IT e software gestionali, impiegati da clienti di varie dimensioni e provenienti da diversi settori, quali l'ambito della sicurezza e della difesa, le telecomunicazioni e la produzione di elettrodomestici. In particolare, LogObject mette a disposizione delle aziende manifatturiere l'applicativo *mLogistics*, che consente la gestione e l'ottimizzazione delle attività logistiche. Grazie a un *real-time scheduling* e al monitoraggio continuo delle risorse, il software permette di gestire la forza lavoro, gli ordini e il flusso di materiali nell'azienda cliente.

I dati analizzati nel progetto derivano dall'applicazione del software in un'azienda leader nella produzione di elettrodomestici in Svizzera. Tramite un'interfaccia sia *mobile* che su computer, diverse risorse interne possono inserire dati relativi a prodotti che abbiano incontrato problematiche di diversa tipologia. In particolare, quando si verifica un guasto in un elettrodomestico prodotto dall'azienda, il cliente chiama l'assistenza e viene reindirizzato a una risorsa del *Back Office*. La risorsa apre un nuovo *task* e raccoglie diverse informazioni circa le caratteristiche del guasto e del cliente, definite come *Call taker data*. Queste informazioni vengono inserite nell'interfaccia *mLogistics* e sono visibili ad un tecnico specializzato (operatore), che viene inviato presso il domicilio del cliente. Nel caso in cui l'operatore abbia con sé tutti i materiali necessari per la riparazione del guasto, il *task* aperto dal *Back Office* può essere considerato chiuso. Nel caso in cui invece siano necessari altri materiali non disponibili al momento della visita a casa del cliente, l'operatore dovrà effettuare multipli viaggi dalla casa del cliente al magazzino e viceversa. Quando questa situazione si verifica, si ha un'elevata inefficienza e il *task* viene identificato come critico. Maggiore è il numero di viaggi necessari per riparare un singolo dispositivo, maggiore è la criticità del *task*. A sua volta, l'operatore inserisce nell'interfaccia software diverse informazioni relative al guasto, quali il codice errore e i materiali portati ad ogni visita presso la residenza del cliente. Tali informazioni sono disponibili solo a posteriori e vengono identificate come *Data reported from the service engineer*.

Lo scopo principale di questa tesi è l'elaborazione di un sistema che permetta, a partire dai *Call taker data*, di prevedere le informazioni relative al codice errore della macchina e alla tipologia di materiali necessari all'operatore per la riparazione del guasto. Questo consentirebbe allo stesso di arrivare al domicilio del cliente già con tutti i materiali necessari alla riparazione del guasto. Tale condizione permetterebbe di ridurre il numero di *task* critici, apportando una serie di vantaggi:

- Riduzione di costi logistici non programmati, dovuti a un'impropria gestione dei trasporti e a mancanza di efficienza.
- Riduzione del consumo di carburante.
- Ottimizzazione del tempo delle risorse e potenziale incremento del numero di *task* eseguiti al giorno.
- Aumento della soddisfazione del cliente grazie a una minore attesa e una ridotta necessità di essere disponibile a casa più volte a causa di multiple visite dell'operatore.

L'obiettivo del progetto può quindi essere sintetizzato come segue:

“Sviluppare un sistema di supporto decisionale per attività di manutenzione di elettrodomestici, prevedendo i materiali necessari alla loro riparazione e riducendo il numero di viaggi necessari dalla casa del cliente ai magazzini dell'azienda e viceversa”.

2. Strategia del progetto

2.1 Roadmap e modalità di svolgimento del progetto

Le analisi sono state realizzate su un set di *Call taker data*, mediante diversi *tool* di implementazione del linguaggio Python (Google Colab, Spyder, Jupyter Notebook). Tale lavoro è stato svolto con un confronto continuo sia con i tutor accademici, sia con il tutor aziendale. In particolare, la conoscenza pregressa del progetto da parte del tutor aziendale ha permesso di validare le conclusioni tratte in fase di esplorazione dei dati e di valutare i risultati finali. L'affiancamento dei tutor accademici ha inoltre dato modo di validare la logica e l'implementazione pratica dei diversi modelli sviluppati. La *roadmap* seguita per il raggiungimento dell'obiettivo è sintetizzata in Fig. 1, che presenta le principali fasi seguite nel progetto e descritte in maggiore dettaglio nelle successive sezioni.

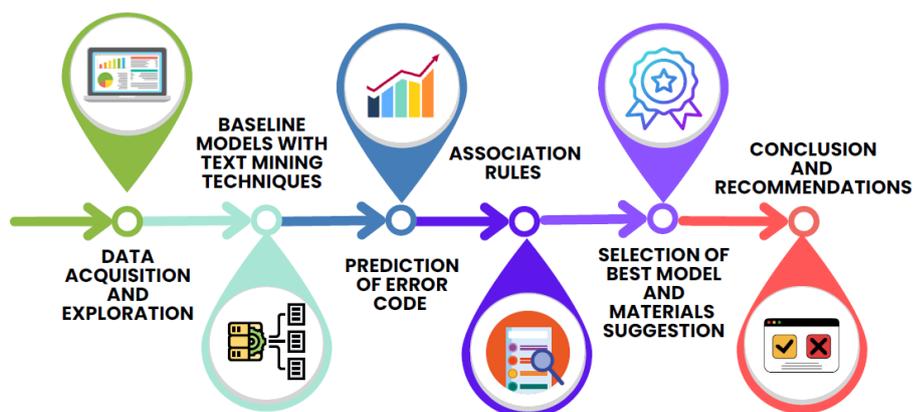


Figura 1: Roadmap seguita per il raggiungimento dell'obiettivo di progetto

2.2 Metriche di valutazione dei modelli

I dati a disposizione sono stati divisi in due set principali: un *training set*, usato come set di dati storici per lo sviluppo dei modelli, e un *test set*, contenente istanze nuove e mai viste dai modelli. La Tab. 1 presenta un gruppo di metriche progettate per valutare il raggiungimento dell'obiettivo da parte dei diversi modelli e calcolate sul *test set*.

Nome metrica	Definizione	Valori possibili
<i>Δ materials</i>	Frazione media di materiali correttamente suggeriti dal modello. Il valore è desiderato il più alto possibile.	0 – 1
<i>Δ visits</i>	Riduzione del numero di visite presso l'abitazione del cliente. Maggiore il suo valore, migliori le performance del modello.	0 – 1
<i>Δ closed</i>	Frazione di <i>task</i> chiusi sul totale dei <i>task</i> presenti nel <i>test set</i> , desiderato quindi il più alto possibile.	0 – 1
<i>Δ critic</i>	Frazione di <i>task</i> critici sul totale di <i>task</i> chiusi grazie al modello. Maggiore il valore, migliori le performance.	0 – 1
<i>Avg Mat</i>	Numero medio di materiali suggeriti dal modello.	0 – ∞
<i>Theta</i>	Numero medio di materiali suggeriti dal modello ma non effettivamente necessari per la riparazione del guasto.	0 – ∞
<i>Perc Theta</i>	Percentuale media di materiali non necessari sul totale dei materiali suggeriti. Minore il suo valore, maggiore la precisione del modello.	0 – 100
<i>Coverage</i>	Percentuale di istanze del <i>test set</i> che hanno avuto suggerimenti dal modello. Il valore è desiderato più alto possibile a parità di performance.	0 – 100

Tabella 1: Definizione delle metriche di valutazione dei modelli

3. Esplorazione e preparazione dei dati

Il *dataset* analizzato raccoglie dati di cinque anni, da settembre 2018 a ottobre 2022, per un totale di 1176645 istanze. Tali osservazioni sono descritte da 30 variabili, più 150 fattori riferiti ai materiali. Il loro significato è stato identificato grazie a tecniche di *data visualization* e tenendo conto delle principali statistiche dei dati. Le conclusioni tratte sono poi state

validate da un confronto con il tutor aziendale. Le variabili del gruppo *call taker data* sono raggruppabili in diverse categorie: quelle geografiche (indicative della localizzazione del cliente), informazioni testuali, variabili temporali relative al momento in cui è stato realizzato il *task* e così via. Tra di esse, figurano alcune di particolare importanza:

- *Task ID* e *Task Report ID*: la prima variabile è un codice identificativo di un *task*, ovvero di tutte le operazioni riferite allo stesso elettrodomestico e allo stesso cliente. Il *Task Report ID* identifica invece ogni singola visita presso lo stesso cliente.
- *Notification Text*: si tratta di una variabile testuale di fondamentale importanza, in quanto contiene la descrizione del tipo di guasto. Essa può essere espressa in tre lingue diverse: francese, tedesco e italiano.
- *Task Type ID*: codice identificativo della tipologia di *task*. Le tipologie possibili sono due, ovvero “riparazione semplice” e “riparazione di fabbrica”.
- *Appliance Master ID*: si tratta di un codice identificativo che indica il tipo di elettrodomestico a cui si riferisce il *task*.
- *Customer Appliance Master Desc*: descrizione dell’elettrodomestico.
- *Plan Date*: rappresenta la data in cui è stato svolto il *task*.

Nel valutare i risultati, è importante anche tenere conto del fatto che il numero di variabili di interesse in input è molto limitato. Per quanto riguarda invece le variabili *Reported from the service engineer*, oggetto di predizione dei modelli, ne figurano innanzi tutto due relative alla tipologia di errore. L’*Error Cause Code* è un codice identificativo del tipo di errore incontrato dalla macchina. La variabile *Error Class Code* rappresenta la medesima informazione in modo più generico: si tratta di un insieme di classi di errore in cui sono raggruppati i diversi *Error Cause Code*. Infine, i dati riportati dall’operatore includono una serie d’informazioni che descrivono i materiali usati per la riparazione del guasto. Tra le variabili relative ai materiali sono di particolare interesse le seguenti:

- *Material X Number*: si tratta di un codice identificativo del tipo di materiale.
- *Material X Desc*: informazione testuale descrittiva del tipo di materiale.

La *X* indicata accanto alla parola “*Material*” può variare tra 1 e 30. Questo significa che, dato un certo valore di *Task Report ID*, se le variabili relative ai materiali hanno valore non nullo per *X* da 1 a 30, in quella specifica visita l’operatore ha impiegato 30 materiali differenti per eseguire il *task*. Nel caso in cui invece le variabili abbiano valore mancante per *X* da 1 a 30, significa che nessun materiale è stato utilizzato. Circa il 42.5% delle istanze del *dataset* non

presentano alcun materiale impiegato per la realizzazione del *task*. Prima di procedere con l'applicazione dei modelli, sono state eseguite le seguenti operazioni di preparazione del *dataset*:

- Identificazione della lingua in cui è espressa la variabile *Notification Text* e suddivisione dei dati in tre gruppi diversi in funzione di tale lingua. I modelli sono poi stati applicati per ragioni computazionali sul solo *dataset* con descrizioni in italiano.
- Modifica della struttura del *dataset* in modo da raggruppare tutte le informazioni relative al medesimo *Task ID* in una sola riga. Le variabili ripetute, come il *Task Report ID* e quelle relative ai materiali, sono state trasformate in liste. Tutti i materiali relativi al medesimo *task*, ma portati in visite diverse, sono quindi stati condensati in una unica variabile *Material List*.
- Operazioni di *preprocessing*, quali eliminazione dei valori mancanti e dei valori inconsistenti, nonché *preprocessing* delle variabili testuali mediante eliminazione di *stopwords*, *lowercasing*, eliminazione della punteggiatura e *tokenizzazione*.

4. Sviluppo di modelli *baseline* basati su tecniche di *text mining*

I primi modelli sviluppati sono definiti come *baseline*, in quanto basati su semplici tecniche di *text mining* e utilizzati come *benchmark* per modelli più sofisticati. La logica del primo modello *baseline* è mostrata in Fig. 2.

4.1 Modello *baseline* basato sulla similarità tra stringhe

Nel primo modello *baseline*, data una certa istanza del *test set*, i dati del *training set* vengono filtrati secondo tre criteri. Le osservazioni risultanti vengono poi sfruttate per effettuare un suggerimento di materiali da utilizzare per l'istanza del *test set* analizzata. I criteri di filtraggio sono i seguenti:

- *Filtro 1*: Vengono selezionate solo le istanze del *training set* aventi lo stesso *Task Type ID* dell'istanza di *test set* analizzata.
- *Filtro 2*: Tra le istanze selezionate dal primo filtro, vengono considerate solo le osservazioni con lo stesso *Appliance Master ID* dell'istanza del *test set* analizzata.
- *Filtro 3*: Viene calcolata la similarità tra *Notification Text* dell'istanza di *test set* e di tutte le osservazioni del *training set* derivanti dal secondo filtro. Solo le istanze con similarità maggiore o uguale a un valore di *threshold* vengono selezionate. I materiali usati per queste istanze sono poi suggeriti per l'istanza di *test set*.

Il terzo filtro si basa dunque sull'andare a calcolare la similarità tra stringhe. Tale valore è stato calcolato in due modi differenti: sia mediante la classe di Python *SequenceMatcher*, sia mediante la più tradizionale tecnica di *Cosine similarity* con *term frequency vectors*. Il *threshold* impiegato nel terzo filtro è stato testato pari a diversi valori: 65%, 75%, 85% e 90%.

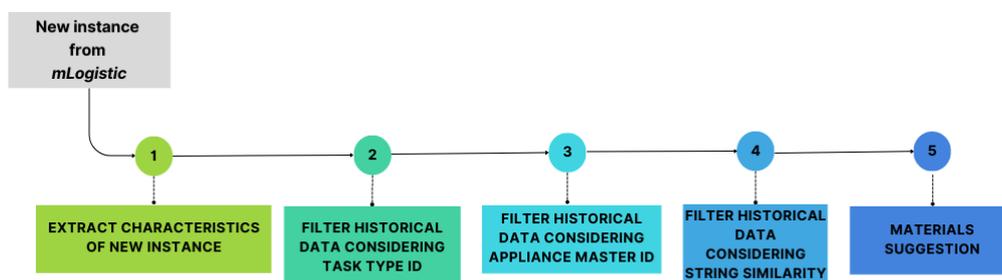


Figura 2: Schema del modello *baseline* basato su similarità tra stringhe

4.2 Modello *baseline* basato sulla *tokenizzazione*

La logica del secondo modello *baseline* è la medesima del primo, ma basata su differenti filtri. Questi ultimi partono dal concetto di *tokenizzazione*, ossia l'estrazione di singoli termini all'interno di una frase. I filtri applicati per ogni istanza del *test set* sono i seguenti:

- *Filtro 1*: Selezione dei dati del *training set* contenenti nelle proprie descrizioni di guasto almeno un termine della variabile *Notification Text* dell'istanza di test.
- *Filtro 2*: Tra le osservazioni risultanti dal primo filtro, selezione di quelle contenenti nelle proprie descrizioni del tipo di macchina almeno un termine della variabile *Customer Appliance Master Desc* dell'istanza di test analizzata.

Nel caso in cui il secondo filtro produca almeno 12 istanze di *training set*, queste vengono filtrate ulteriormente considerando l'ID dell'*appliance*. I materiali più frequentemente impiegati nelle istanze di *training set* risultanti dai vari filtri vengono poi suggeriti come lista di materiali da usare per il *task* del *test set* analizzato.

4.3 Varianti del modello *baseline* basato su similarità tra stringhe

Il modello basato sulla similarità è stato testato anche con le seguenti modifiche:

- Introduzione di un ulteriore filtro basato sull'anno in cui è stato effettuato il *task*. Questa variante è stata attuata in modo da verificare se, eventualmente, in anni diversi gli stessi materiali fossero indicati con codici diversi. Tuttavia, il filtro non ha apportato cambiamenti significativi.
- Aggiunta di un ulteriore filtro basato sulla similarità tra stringhe, considerando anche la descrizione del tipo di elettrodomestico. Anche in questo caso, i risultati non sono cambiati in modo significativo rispetto alla versione "base".

- Introduzione di una tecnica di *text mining* più sofisticata per determinare la similarità tra le descrizioni dei guasti. In particolare, questa versione si è basata sull'uso di BERT per la codifica delle frasi in vettori numerici, chiamati *embeddings*. BERT è un modello pre-addestrato, in grado di produrre rappresentazioni del testo semanticamente significative e non basate sulla semplice frequenza dei termini. Una volta rappresentate le frasi mediante BERT, la similarità è poi stata calcolata nuovamente mediante *Cosine similarity*. Questa particolare versione ha prodotto i risultati migliori ed è stata poi confrontata con i modelli più avanzati. I valori delle metriche per tale variante figurano nella Tab. 2 sotto il nome di "Modello 1".

5. Sviluppo di un modello basato su tecniche di *machine learning*

Il secondo modello è stato sviluppato per soddisfare la richiesta del *team* di R&S di creare un modello che implementasse un processo a due step. Il primo consiste nell'andare a prevedere l'*Error Cause Code* a partire da una serie di *Call taker data*. Nel secondo step vengono invece suggeriti i materiali da usare per il *task* in base all'errore previsto. Dato che i codici di errore distinti analizzati sono 50, la previsione dell'errore rappresenta un problema di classificazione di tipo multi-classe.

Il *dataset* è inoltre caratterizzato da un forte sbilanciamento tra le classi. Infatti, le due maggioritarie da sole costituiscono circa il 55% del totale dei dati. Queste caratteristiche affliggono in modo significativo le performance dei modelli di classificazione.

Per addestrare i modelli di previsione dell'errore, i dati del *training set* sono stati ulteriormente suddivisi in modo da ottenere un *training set* finale effettivo e un *validation set*, usato per testare i modelli in modo da poter definire i valori ottimali dei parametri dei classificatori e valutare la necessità di introdurre o meno un'operazione di *oversampling*, visto l'elevato sbilanciamento delle classi. I classificatori testati per la previsione dell'*Error Cause Code* sono stati tre. Innanzi tutto, è stato testato il modello *Random Forest*, che con la migliore combinazione di parametri e *oversampling*, ha generato un *F-score* pari al 41.93%. Il secondo modello testato, *Support Vector Machines*, impiegato insieme all'operazione di *oversampling* ha generato un *F-score* massimo pari al 42.30%. Infine, l'ultimo classificatore testato è stato *SAMME AdaBoost*. Tuttavia, quest'ultimo ha mostrato risultati particolarmente scarsi, con *F-score* non superiori a 21.76%. Data la difficoltà dei modelli di classificazione nel gestire la configurazione di classi, è stato applicato un algoritmo di *clustering* gerarchico che permettesse di raggruppare codici di errore simili per materiali

impiegati e prevedere gruppi di errori anziché singoli codici. Il *clustering* è stato applicato a una matrice composta da righe rappresentanti ognuna un codice errore, e con colonne corrispondenti a diversi codici materiali.

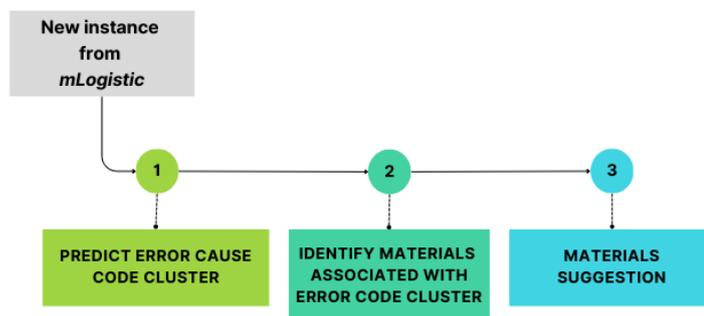


Figura 3: Schema del modello basato su classificazione e *clustering*

Le componenti di tale matrice rappresentano invece la frequenza con cui il materiale della colonna corrispondente viene usato per il tipo di errore della riga corrispondente. Questo ha permesso anche di creare un modo per suggerire in maniera diretta i materiali in base al cluster previsto. Il *clustering* ha portato alla generazione di 26 gruppi di codice errore, per i quali è stato nuovamente addestrato il classificatore *Support Vector Machines*, ossia il più performante nel prevedere i singoli codici. La previsione sui *cluster* mediante tale modello ha portato a un miglioramento rispetto alla classificazione sui singoli codici, con un *F-score* pari al 49.99%. I risultati legati al suggerimento dei materiali sono riportati in Tab. 2 sotto il nome di “Modello 2”, mentre la logica generale seguita è sintetizzata in Fig. 3.

6. Sviluppo di un modello basato su *frequent patterns mining*

I modelli di *frequent patterns mining* si basano sulla ricerca di relazioni frequenti e ricorrenti all’interno di un set di dati. In particolare, il modello sviluppato si basa sulla ricerca di *frequent itemset*, ovvero di valori che figurano di frequente insieme nelle righe del *dataset*. Tali oggetti sono stati identificati per il *training set* mediante un algoritmo d’identificazione di *frequent itemsets* definito *FP-Growth*. Sulla base dei *frequent itemsets*, è stato poi costruito un set di regole associative del tipo: $A \Rightarrow B$. In tale regola, l’antecedente A è rappresentato da combinazioni di possibili valori di *Appliance Master ID*, *Task Type ID*, *Plan Date*. B rappresenta invece il conseguente della regola e in questo caso è stato identificato con i valori assunti dalle variabili *Material X Number*. Tali regole associative sono state filtrate in funzione di alcune metriche di qualità, per poi essere impiegate nel suggerimento di materiali da impiegare per le diverse istanze del *test set*, secondo la logica di Fig. 4.

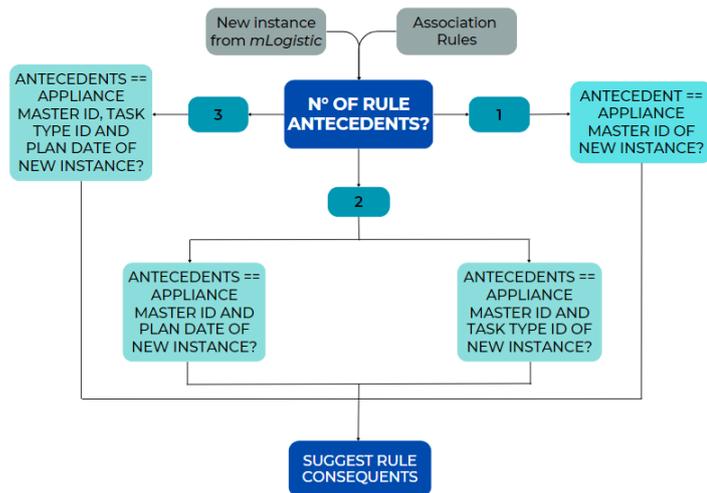


Figura 4: Schema di funzionamento del suggerimento di materiali basato su *association rules*

Anche per questo modello sono state testate alcune varianti con tali modifiche:

- Sostituzione della data di esecuzione del *task* tra gli antecedenti delle regole con il solo anno di esecuzione del *task*.
- Generazione iterativa di regole, eliminando via via dai conseguenti i materiali più utilizzati, per favorire il suggerimento di quelli meno frequenti.
- Modifica dei criteri di qualità usati per selezionare le regole. Questa variante ha ottenuto i medesimi risultati della precedente, nonché i migliori tra le diverse versioni. I risultati sono riportati in Tab. 2 sotto il nome “Modello 3”.
- Aggiunta d’informazioni testuali negli antecedenti delle regole, grazie all’uso del modello *keyBERT* per l’estrazione di parole chiave dalla descrizione del guasto.

7. Risultati relativi al suggerimento dei materiali

Tab. 2 riporta i valori delle metriche ottenute per le migliori versioni di ogni modello. Non figurano *Theta* e *Avg Mat*, in quanto funzionali al calcolo di *Perc Theta*. Il Modello 3 ha un’elevata precisione, dato il valore più basso di *Perc Theta*. Tuttavia, tale caratteristica è limitata a un piccolo set di osservazioni, data la ridotta *coverage*. Questo affligge gli altri risultati del modello. Il Modello 2 ha il vantaggio di avere una copertura totale del *dataset*, tuttavia ha scarsa precisione: il primo aspetto passa in secondo piano rispetto agli obiettivi del progetto. Inoltre, dato che nel dataset iniziale il 42.5% dei *task* non richiede alcun materiale, questo significa che la *coverage* massima non è statisticamente richiesta. In termini delle altre metriche, il Modello 1 presenta le migliori *performance*. In particolare, grazie ai suggerimenti di tale modello, la frazione media di visite evitate è del 39%. Inoltre, il 43% dei *task* possono essere chiusi con una singola visita e tra questi il 12% sono *task* a

elevata criticità. Grazie anche al compromesso ottimale tra metriche e *coverage*, il Modello 1 è stato quindi identificato come il migliore per il raggiungimento degli obiettivi del progetto.

Modello	Perc Theta	$\Delta materials$	$\Delta visits$	$\Delta closed$	$\Delta critic$	Coverage
Modello 1	67.11 %	0.44	0.39	0.43	0.12	37.21 %
Modello 2	97.94 %	0.14	0.09	0.11	0.08	100 %
Modello 3	37.23 %	0.061	0.039	0.037	0.05	12.62 %

Tabella 2: Risultati relativi al suggerimento di materiali per la migliore versione di ogni modello

8. Conclusioni e proposte di miglioramento

Il progetto ha condotto con successo all'identificazione di un modello in grado di aumentare l'efficienza delle attività di manutenzione, come dimostrato dai valori riportati in Tab. 2. Considerando che un numero ridotto di ricerche nell'applicazione di tecniche di *data mining* ai contesti manifatturieri riguarda l'ambito della manutenzione e della riparazione, l'importanza di tali risultati è ulteriore. Inoltre, il Modello 1 ha il vantaggio di essere un sistema generalizzabile e di semplice implementazione, sia in vari contesti che in diversi linguaggi di programmazione. L'unico svantaggio nell'uso di BERT è rappresentato dagli elevati tempi computazionali per la produzione degli *embeddings*, comunque evitabili processando i dati storici periodicamente e non ad ogni arrivo di una nuova istanza. Sono infine stati identificati alcuni possibili miglioramenti per futuri sviluppi del progetto:

- Variazione della modalità di raccolta dei dati testuali. In particolare, inserendo un menù a tendina con descrizioni standardizzate e prestabilite dei guasti più frequenti, si garantirebbe una maggiore consistenza rispetto all'inserimento manuale e incrementerebbe la qualità delle informazioni testuali.
- Possibilità di effettuare un *fine-tuning* di BERT, adattandolo a identificare eventuali errori ortografici e al linguaggio tecnico relativo alla riparazione di elettrodomestici.
- Possibilità di raggruppare i materiali per similarità, suggerendo agli operatori non singoli materiali, ma liste di codici tra cui scegliere. Una tecnica di raggruppamento basilare è stata testata su uno dei modelli baseline, portando a un miglioramento in termini di $\Delta closed$ e di $\Delta critic$.
- *Re-focus* dell'obiettivo di progetto, tenendo conto del costo dei materiali suggeriti ma non effettivamente utilizzati. Un'attenta analisi di tali costi, unita a quelli dovuti a viaggi multipli e al tempo mal impiegato dell'operatore, permetterebbe d'identificare un obiettivo di progetto più preciso e mirato.