



## POLITECNICO DI BARI

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING  
Master Degree in Data Science

Dissertation in Artificial Intelligence

# Un cruscotto personalizzato per l'agente assicurativo

Supervisore  
Prof. Eng. Fedelucio Narducci

Candidato  
Mauro Di Liddo

Anno Accademico 2023 - 2024

## Sommario

1.	Introduzione al problema .....	6
2.	Analisi funzionale .....	8
2.1.	Profilo assicurativo.....	8
2.2.	Coperture assicurative .....	10
2.3.	Coperture selezionate.....	11
3.	I sistemi di raccomandazione.....	12
3.1.	Sistemi basati sul contenuto .....	13
3.2.	Sistemi collaborativi .....	14
3.2.1.	Sistemi memory Based (neighborhood based) .....	15
3.2.2.	Sistemi model based.....	17
3.3.	Sistemi ibridi.....	19
3.4.	Valutazione di un sistema .....	19
3.5.	Considerazioni in ambito assicurativo .....	19
4.	Soluzione implementata .....	21
4.1.	Acquisizione dati .....	21
4.2.	Analisi dei dati.....	24
4.3.	Valutazione del modello .....	26
4.4.	Sistema di raccomandazione .....	28
4.4.1.	Approccio collaborativo .....	28
4.4.2.	Addestramento classificatore .....	30
4.4.3.	Approccio ibrido .....	31
5.	Architettura.....	38
6.	Conclusioni e sviluppi futuri.....	49

*Ringrazio:*

*Tiziana Loporchio ed Alessandra Boccaperta per avermi dato la possibilità di frequentare questo master.*

*Gianfranco Cassano e Carlo Aprile per avermi sostenuto e seguito durante il tirocinio e la tesi.*

*Tutti i colleghi per i suggerimenti e le indicazioni ricevute durante quest'anno.*

*A mamma, papà e Pallino.*

*Considerate la vostra semenza:  
fatti non foste a viver come bruti,  
ma per seguir virtute e canoscenza.  
(Divina Commedia, Inferno XXVI).*

# 1. Introduzione al problema

Il progetto presentato prevede l'analisi del sistema emissivo polizze di un cliente assicurativo di **Fincons Group S.p.A.**, lavorando sul sistema in questione ho notato delle peculiarità del sistema che andrebbero migliorate.

L'attuale sistema emissivo delle polizze assicurative risulta essere farraginoso, di poco impatto e anonimo. L'immagine mostra la pagina utilizzata dagli agenti per l'emissione delle polizze.

The screenshot shows a web-based application interface for insurance agents. At the top, there's a header with the title 'Dat Assicurazioni' and a user profile for 'Giulia Bianchi'. On the left, a sidebar menu includes 'Cambi', 'Polizze', 'Prevention', and 'Contabilità'. The main content area displays a client profile for 'Rossi Mario' with details like 'CODICE FISCALE: RSSMRA54P24F284G', 'DATA DI NASCITA: 24/09/1954', 'ETA': 69, 'INDIRIZZO: VIA A.MANZONI, 14 - 00157 ROMA (RM)', and 'PROFESSIONE: IMPIEGATO'. Below this, there are several sections with icons and buttons: 'Viaggi' (Flight, Train, Car, Bus), 'Attività Lavorativa' (Agriculture, Breeding, Industrial Enterprises), 'Velcoll' (Auto, Moto, Imbarcazioni, Auto d'affari), 'Famiglia' (Pensione, Animali Domestico), 'Salute' (Vasta Specialistica, Odontoiatrica, Grandi Interventi), and 'Patrimonio' (Case, Beni di Lusso). The bottom of the page has a footer with the text 'Pagina iniziale emissione polizze'.

L'agente assicurativo non ha a disposizione una panoramica esauriente del cliente che sta gestendo; per visualizzare il profilo del cliente deve aprire la pagina del profilo completo, senza avere la possibilità di visualizzare contemporaneamente le sue informazioni e le coperture assicurative che può proporgli. Nell'immagine che segue vengono mostrati i dati del cliente a disposizione dell'agente nella pagina di emissione.

This screenshot shows a detailed view of the client profile for 'Rossi Mario'. It includes the same basic information as the previous page: 'CODICE FISCALE: RSSMRA54P24F284G', 'DATA DI NASCITA: 24/09/1954', 'ETA': 69, 'INDIRIZZO: VIA A.MANZONI, 14 - 00157 ROMA (RM)', and 'PROFESSIONE: IMPIEGATO'. Below this, there's a section labeled 'Dati Cliente' which likely contains more specific client details. The overall layout is clean and organized, providing all necessary information at once.

L'agente assicurativo visualizza le coperture assicurative che è possibile vendere al cliente suddivise per tipologia prodotto. Le coperture assicurative raggruppate secondo questo criterio non sono personalizzate; quindi, l'agente non ha alcun supporto riguardo le coperture assicurative che sarebbe opportuno proporre al cliente. Se le coperture assicurative di una tipologia sono numerose non è possibile visualizzarle tutte contemporaneamente nella stessa schermata. Nell'immagine vengono mostrate le coperture assicurative presenti in pagina secondo il raggruppamento descritto.

The screenshot displays a user interface for insurance coverage selection. It is organized into several sections:

- Viaggi**: Contains buttons for **Bagagli**, **Voli**, **Treni**, and **Crociere**.
- Attività Lavorativa**: Contains buttons for **Agricoltura**, **Allevamento**, and **Impianti Industriali**.
- Veicoli**: Contains buttons for **Auto**, **Moto**, **Imbarcazioni**, and **Auto d'Occasione**.
- Famiglia**: Contains buttons for **Familiare** and **Animale Domestico**.
- Salute**: Contains buttons for **Visite Specialistiche**, **Odontoiatria**, and **Grandi Interventi**.
- Patrimonio**: Contains buttons for **Casa** and **Beni di lusso**.

At the bottom center of the interface is the text **Elenco coperture assicurative**.

La pagina descritta è stata realizzata utilizzando in parte tecnologie legacy, quindi ogni volta che l'utente interagisce con la pagina essa viene ricaricata, senza garantire una buona interazione fornendo all'utente un'esperienza di navigazione limitata e deludente.

Dall'analisi eseguita sono emerse le problematiche riguardo l'utilizzo dell'applicativo:

- Durante l'emissione delle polizze assicurative, è possibile visualizzare solo una minima parte dei dati relativi al profilo cliente.
- Le coperture assicurative sono visualizzate indipendentemente dal cliente selezionato.
- Le coperture assicurative sono raggruppate in base alla tipologia di prodotto e non in base all'interesse del cliente.
- L'applicazione fornisce una scarsa interattività ed un'esperienza utente deludente.

## 2. Analisi funzionale

Per risolvere i problemi descritti nel capitolo precedente si propone la realizzazione di una nuova dashboard.

La sua progettazione ruota intorno al cliente ed è stata pensata per essere utilizzata dagli agenti assicurativi. Sono state previste tre sezioni:

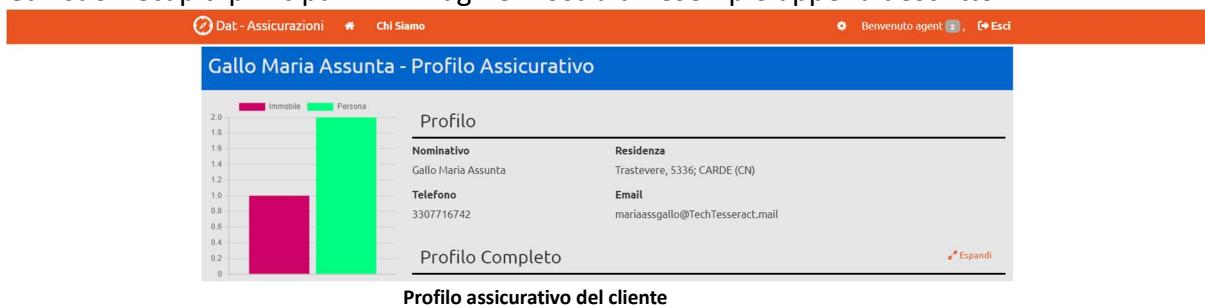
- Profilo assicurativo del cliente
- Sezione coperture assicurative
- Sezione beni selezionati

Il cruscotto progettato avrà l'aspetto mostrato nell'immagine:



### 2.1. Profilo assicurativo

Il profilo del cliente è stato pensato per rendere disponibili all'agente sempre le informazioni fondamentali, su richiesta le altre informazioni. L'agente avrà subito a disposizione un grafico riassuntivo con leggenda delle coperture assicurative, il nominativo, ed i suoi recapiti principali. L'immagine mostra un esempio appena descritto.



Nella schermata visualizzata sono mostrati:

- Il nominativo del cliente nell'intestazione della pagina.

- Sulla sinistra un grafico riassuntivo con leggenda delle coperture attive. Il colore delle coperture presenti nel grafico corrisponde al colore definito per la categoria di polizza.
- Sulla destra i dati essenziali del cliente, nominativo, indirizzo principale o di residenza, il telefono principale, l'indirizzo email principale.

Cliccando sul pulsante *espandi* presente in basso a destra verrà visualizzato il profilo completo del cliente. È prevista una scheda per ogni tipologia di dato *anagrafica*, *indirizzo*, *recapiti*, professione; cliccando su di una delle schede presenti verranno visualizzati i dati corrispondenti.

The screenshot shows the 'Gallo Maria Assunta - Profilo Assicurativo' page. At the top left is a bar chart titled '2023' with two bars: one red ('Immobile') at approximately 0.9 and one green ('Persona') at approximately 2.0. To the right of the chart is a 'Profilo' section with 'Nominativo' (Gallo Maria Assunta), 'Residenza' (Trastevere, 5336; CARDE (CN)), 'Telefono' (3307716742), and 'Email' (mariaassgallo@TechTesseract.mail). Below this is a 'Profilo Completo' section with tabs for 'Anagrafica' (selected) and 'Indirizzi', 'Recapiti', and 'Professione'. Under 'Anagrafica', details include 'Nominativo' (Gallo Maria Assunta), 'Sesso' (Donna), 'Stato civile' (Single), 'Numero figli' (1), 'Data di nascita' (venerdì 24 agosto 1934), and 'Luogo di nascita' (63839 - SERVIGLIANO (FM)). A 'Comprimi' button is located at the top right of the 'Profilo Completo' section. Below the main content is a bold 'Profilo completo cliente' heading.

Cliccando sul pulsante *espandi* presente a destra nella sezione *coperture assicurative* verrà visualizzato un report riassuntivo contenente le polizze stipulate dal cliente. L'immagine mostra un esempio del report appena descritto:

The screenshot shows the same 'Gallo Maria Assunta - Profilo Assicurativo' page with the 'Coperture Attive' section expanded. It includes a bar chart and a table of policies. The table has columns: Identificativo Polizza, Tipologia Polizza, Data Emissione, Data Scadenza, and Premio Totale. The data is as follows:

Identificativo Polizza	Tipologia Polizza	Data Emissione	Data Scadenza	Premio Totale
IM002-GFB4OPQ005-1	GLOBALE FABBRICATI	13/03/2024	08/03/2025	1.747,00 €
PP003-INF5RST006-1	INFORTUNI	19/03/2024	14/03/2025	1.091,91 €
PP003-INF5RST006-2	INFORTUNI	16/02/2024	10/02/2025	3.804,04 €

Below the table, it says '0 selected / 3 total'. A 'Coperture assicurative' heading is at the bottom of the expanded section.

Nel report per ogni copertura visualizzata ci sono i dati:

- Codice univoco polizza
- Tipologia polizza
- Data stipulazione contratto
- Data scadenza contratto
- Importo premio totale

Per ogni copertura presente nella griglia c'è un pulsante che espande la riga e visualizza il dettaglio della polizza specificando:

- Dettaglio dei beni assicurati.
- Elenco garanzie selezionate.

Coperture Attive					 Comprimi
Identificativo Polizza	Tipologia Polizza	Data Emissione	Data Scadenza	Premio Totale	
<a href="#">IM002-GFB4OPQ005-1</a>	GLOBALE FABBRICATI	13/03/2024	08/03/2025	1.747,00 €	
<a href="#">PP003-INF5RST006-1</a>	INFORTUNI	19/03/2024	14/03/2025	1.091,91 €	
<b>Descrizione beni assicurati:</b>					
Nominativo: Martini Giuseppe Maria - Nato il: 01/04/2004 - Relazione: Figlio/a - Infortunio vita privata - Infortunio sfera professionale Nominativo: Galli Lorenzo - Nato il: 10/09/1996 - Relazione: Fratello/Sorella					
<b>Garanzie selezionate:</b>					
GR00026 - Invalidità permanente totale o parziale - garanzia obbligatoria GR00028 - Spese mediche per visite specialistiche					
<a href="#">PP003-INF5RST006-2</a>	INFORTUNI	16/02/2024	10/02/2025	3.804,04 €	
1 selected / 3 total					

#### Dettaglio coperture assicurative attive

La sezione descritta permette all'agente di avere una panoramica dell'attuale situazione assicurativa del cliente.

## 2.2. Coperture assicurative

La seconda parte della dashboard è incentrata sulle coperture assicurative che verranno proposte al cliente raggruppate in:

- **Coperture suggerite:** sono le coperture assicurative che vengono suggerite all'agente per proporle direttamente al cliente.
- **Coperture più vendute:** sono le coperture più vendute calcolate rispetto ad una statistica eseguita in base alle vendite dell'anno precedente a quello in corso.
- **Altre coperture:** sono le coperture escluse dai precedenti raggruppamenti.

Ogni copertura è rappresentata tramite una *card*, contenente i dati:

- Icona del prodotto
- Nome del prodotto
- Categoria del prodotto
- Descrizione del prodotto
- Spunta attraverso la quale è possibile selezionare o deselectonare il prodotto

Il colore utilizzato per rappresentare l'icona, il nome e la categoria del prodotto è il colore definito per la categoria.

Le coperture presenti all'interno della sezione **coperture suggerite** se fanno parte delle due coperture più vendute; alla copertura suggerita corrispondente alla copertura maggiormente venduta viene aggiunta un'icona che rappresenta una coccarda.

 <b>R.C. DIVERSI - Linea Prodotto: Responsabilità Civile</b> Responsabilità civile per danni riguardo eventi e manifestazioni sportive agonistiche ed amatoriali <input type="checkbox"/> Spunta la casella per aggiungere il bene	 <b>A.R.D. - Linea Prodotto: Responsabilità Civile</b> Assicurazione per la responsabilità civile per danni arrecati a terzi da parte di animali domestici <input type="checkbox"/> Spunta la casella per aggiungere il bene
---	---

Card relative alle coperture assicurative

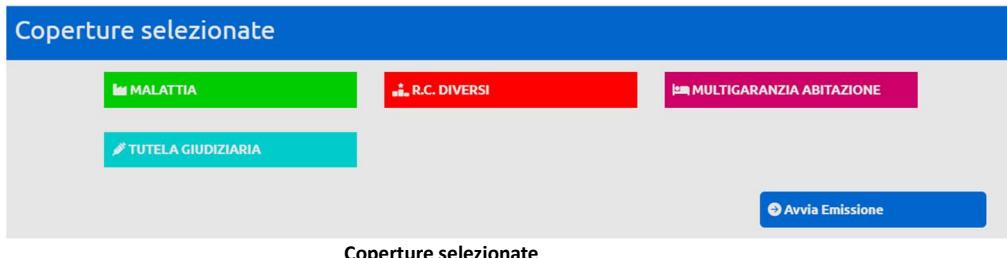
## 2.3. Coperture selezionate

Il terzo raggruppamento presente in fondo alla pagina può essere considerato una bussola per l'utente.

Essa contiene dei segnaposti uno per ogni copertura selezionata, ogni segnaposto contiene:

- Icona del prodotto
- Nome del prodotto

Per rappresentare il colore di sfondo è stato utilizzato il colore definito per la categoria.



### 3. I sistemi di raccomandazione

L'obiettivo di un sistema di raccomandazione<sup>i</sup> consiste nel generare suggerimenti significativi per l'utente del sistema che possano essere di suo interesse. *I libri suggeriti da Amazon oppure i film su Netflix, rappresentano un chiaro esempio del funzionamento di un sistema di raccomandazione.* L'architettura del sistema dipende dal dominio applicativo, dai dati disponibili e dalle modalità con le quali l'utente interagisce con l'applicativo. *Su Netflix dopo aver visto un film è possibile esprimere una valutazione di gradimento.* Le valutazioni che esprime l'utente vengono memorizzate all'interno di una matrice *users-items*.

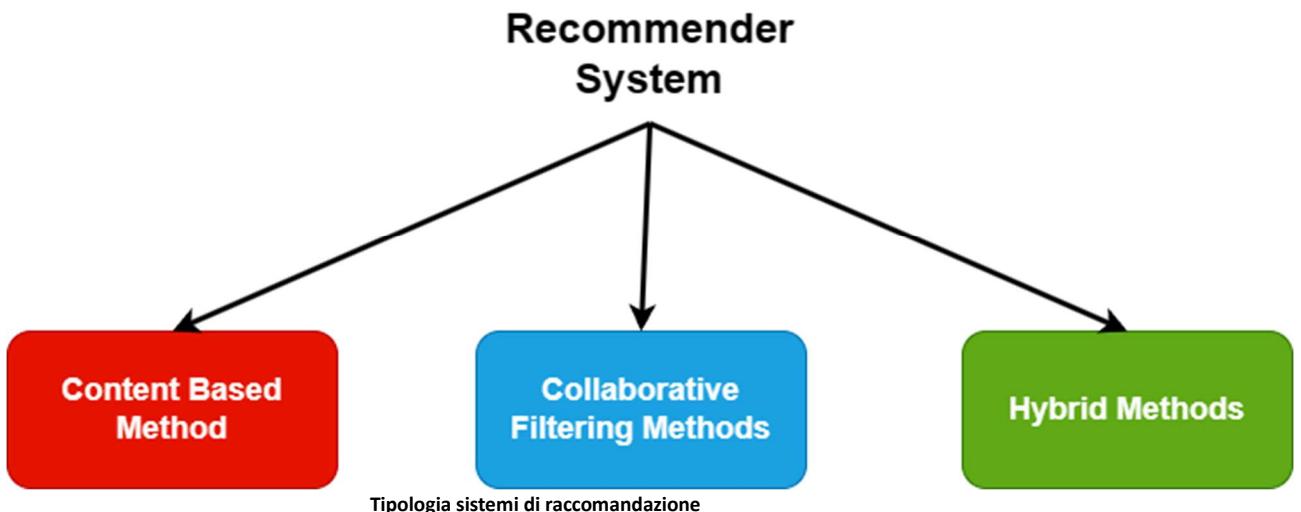
	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$	$i_6$
$u_1$		5			3	
$u_2$	1			3		4
$u_3$		2			2	
$u_4$	4			3		
$u_5$		4	3			5

Matrice Users-Items

La matrice users-items rappresenta le preferenze degli utenti ( $n$  in questo caso 5) per gli items del sistema ( $m$  in questo caso 6). Ogni cella  $r_{ij}$  con  $i$  che va da 1 a 5 e  $j$  che va da 1 a 6 indica la valutazione dell'utente ( $i$ ) per l'item ( $j$ ). La matrice users-items è una matrice sparsa, l'obiettivo di un sistema di raccomandazione è calcolare la valutazione di un generico utente per gli items sui quali non ha espresso alcuna valutazione.

Ci sono diversi approcci ai sistemi di raccomandazione che si possono classificare in:

- Content based recommending: nei sistemi basati sul contenuto gli *items* che vengono suggeriti all'utente hanno un contenuto simile rispetto a quelli a cui l'utente in passato ha dato una valutazione positiva oppure ha acquistato.
- Collaborative Filtering Methods: nei sistemi collaborativi gli *items* suggeriti all'*utente* sono basati sulle valutazioni complessive degli utenti.
- Hybrid approaches: nei metodi ibridi vengono combinate le strategie viste sia per l'approccio basato sul contenuto che quelle basate sui sistemi collaborativi.



### 3.1. Sistemi basati sul contenuto

I sistemi di raccomandazione basati sul contenuto prevedono la creazione di una base di conoscenza creata in base agli elementi che si vogliono proporre al cliente.

Un esempio di base di conoscenza *per una videoteca* potrebbe essere definito come:

- Titolo film
- Descrizione trama
- Cast
- Regista
- Genere

I campi proposti rappresentano un sotto insieme di una possibile base di conoscenza reale; ogni campo a sua volta può essere collegato ad altre basi di conoscenza. Il genere potrebbe essere descritto come:

- Titolo genere
- Descrizione genere
- Generi collegati
- Registi che seguono il genere
- Film del genere

La base di conoscenza creata verrà rappresentata come un grafo, dove i nodi rappresentano le entità ed le associazioni rappresentano la relazione che intercorre tra le entità collegate.

Dopo aver creato e popolato la base di conoscenza bisogna misurare la similarità tra gli elementi; essa viene calcolata utilizzando il **TF-IDF**<sup>ii</sup>, il TF-IDF è costituito dal *TF* - *Term Frequency* e dall'*IDF* - *Inverse Document Frequency*.

Il **Term Frequency** di un termine viene calcolato come il rapporto tra *"il numero di volte che quel termine compare all'interno di un documento"* ed *"il numero totale dei termini presenti all'interno del documento"*.

Numero di occorrenze di un termine all'interno di un documento

$$TF = \frac{\text{Numero di occorrenze di un termine all'interno di un documento}}{\text{Numero totale dei termini presenti nel documento}}$$

Formula per il calcolo di TF – *Term Frequency*

L'**Inverse Document Frequency** di un termine viene calcolato come il *logaritmo* del rapporto tra “*il numero totale dei documenti*” ed “*il numero di documenti nei quali compare il termine analizzato*”.

$$\text{IDF} = \log \left( \frac{\text{Numero totale dei documenti}}{\text{Numero dei documenti nel quale compare il termine analizzato}} \right)$$

Formula per il calcolo di IDF – Inverse Document Frequency

Il **TF-IDF** viene calcolato come il prodotto degli elementi presentati.

$$\text{TF-IDF} = \text{TF} * \text{IDF}$$

Formula per il calcolo di TF-IDF

L'utilizzo del TF-IDF permette di non dover eliminare le “*stop words*<sup>iii</sup>” in quanto il TF-IDF calcolato per esse risulta 0 e quindi non rilevante.

Le raccomandazioni calcolate utilizzando l'approccio *Content Based Methods* prevedono di calcolare la similarità tra le coperture sottoscritte dal cliente e le coperture presenti all'interno della base di conoscenza.

Per calcolare la similarità è necessario rappresentare la base di conoscenza in forma numerica; per forma numerica s'intende creare un vettore relativo ad ogni copertura assicurativa presente nella base di conoscenza. Ogni vettore è costituito dalle caratteristiche individuate della copertura assicurativa rappresentata utilizzando gli **embeddings**<sup>iv</sup>, la similarità viene misurata calcolando il coseno<sup>v</sup> dell'angolo creato fra l'embedding relativo alla copertura assicurativa sottoscritta dal cliente e quelle presente nella base di conoscenza, se il risultato è “1” rappresenta la massima similarità (*tendono ad essere sovrapposti*) mentre se è “0” rappresenta la minima similarità (*tendono ad essere ortogonali*).

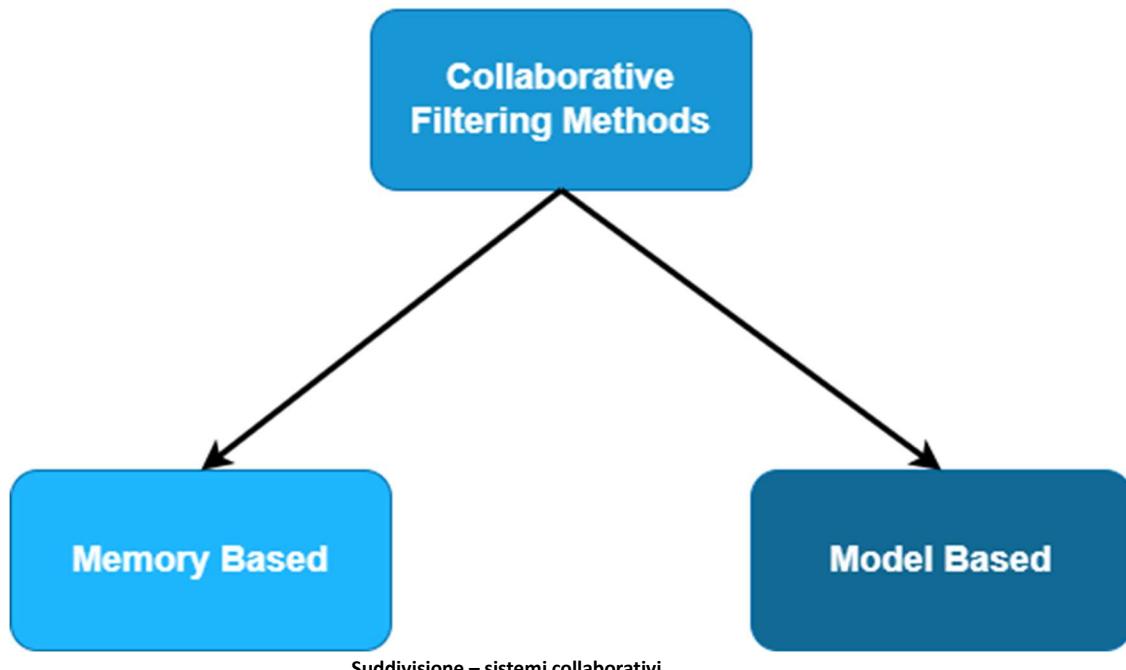
### 3.2. Sistemi collaborativi

I sistemi collaborativi lavorano sulla collezione delle preferenze sugli items indicate dagli utenti per un determinato dominio applicativo, esplorando le similarità presenti fra le valutazioni esistenti. *Per fare un confronto nella vita reale i sistemi collaborativi si basano sul concetto “intuitivo” di passaparola, es. una persona chiede ai propri amici o conoscenti cosa pensano dell’ultimo film uscito al cinema, oppure l’indicazione su quale ristorante scegliere.*

I sistemi collaborativi possono essere suddivisi in due ulteriori categorie:

- Metodo basato sul vicinato (*neighborhood based*) chiamato anche metodo basato sulla memoria (*memory based method*).
- Metodo basato sul modello.

## Collaborative Filtering Sub-divided



### 3.2.1. Sistemi memory Based (neighborhood based)

Le tecniche dei sistemi basati sul vicinato prevedono di calcolare la similarità dell’utente attivo utilizzando un sottoinsieme di utenti. Viene calcolata una media pesata delle loro valutazioni ed utilizzata per la stima della predizione relativa all’utente attivo. I passaggi di questo approccio possono essere schematizzati nell’algoritmo:

1. Assegnare un peso a tutti gli utenti rispettando la similarità con l’utente attivo.
2. Seleziona  $k$ -utenti che hanno la similarità più alta con l’utente attivo, *comunemente chiamati vicinato (neighborhood)*.
3. Calcolare la predizione attraverso una media ponderata tra le valutazioni del vicinato.

Per quanto riguarda il passaggio (1), si definisce  $w_{a,u}$  la misura di similarità tra l’utente attivo  $a$  ed un utente generico  $u$ . La formula più diffusa per il calcolo della similarità è il *coefficiente di correlazione di Pearson*.

$$w_{a,u} = \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r}_a)^2 \sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}}$$

Formula per il calcolo del Coefficiente di correlazione di Pearson tra utente  $a$  ed utente  $u$

Gli elementi presenti nel coefficiente di correlazione di Pearson sono identificati:

- $I$ : rappresenta l'insieme delle preferenze definite dagli utenti.
- $r_{u,i}$ : è la valutazione dell'utente  $u$  rispetto all'item  $i$ .
- $r_{a,i}$ : è la valutazione dell'utente attivo  $a$  rispetto all'item  $i$ .
- $\bar{r}_u$ : indica la media delle valutazioni eseguite dall'utente  $u$ .
- $\bar{r}_a$ : indica la media delle valutazioni eseguite dall'utente attivo  $a$ .

Nel passaggio (3) le predizioni vengono calcolate come una media ponderata dello scarto relativo alle valutazioni dei vicini rispetto alla media delle loro valutazioni.

$$p_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

**Formula per il calcolo delle predizioni**

La formula relativa alle predizioni è definita come segue:

- $p_{a,i}$ : è la previsione per l'utente  $a$  relativa all'elemento  $i$ .
- $w_{a,u}$ : è la similarità tra l'utente  $a$  e l'utente  $u$ .
- $K$ : è il vicinato, formato dagli utenti simili.

La similarità calcolata attraverso il coefficiente di correlazione di Pearson misura la distanza tra due variabili linearmente dipendenti. Una possibile alternativa al coefficiente di correlazione di Pearson è il calcolo della similarità utilizzando il coseno.

$$w_{a,u} = \cos(\vec{r}_a, \vec{r}_u) = \frac{\vec{r}_a \cdot \vec{r}_u}{\|\vec{r}_a\|_2 \times \|\vec{r}_u\|_2} = \frac{\sum_{i=1}^m r_{a,i} r_{u,i}}{\sqrt{\sum_{i=1}^m r_{a,i}^2} \sqrt{\sum_{i=1}^m r_{u,i}^2}}$$

**Formula per il calcolo del coseno tra vettori**

Studi empirici<sup>vi</sup> hanno dimostrato che il coefficiente di correlazione di Pearson offre prestazioni migliori rispetto alla similarità del coseno. Altre misure di somiglianza usate nella letteratura includono la correlazione di Spearman, la correlazione di Kendall, le differenze quadrate medie, l'entropia e la similarità del coseno aggiustata.

L'approccio basato sul vicinato risulta essere poco scalabile in quanto la ricerca di utenti simili ha un'elevata complessità computazionale. Linden<sup>vii</sup> propone un approccio *collaborativo item-to-item*, in questo approccio vengono accoppiati gli item simili tra loro partendo dagli item valutati dall'utente. Seguendo questo approccio il coefficiente di correlazione di Pearson viene modificato ottenendo:

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2} \sqrt{\sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}$$

Formula del Coefficiente di correlazione di Pearson relativo all'approccio *item to item*

Gli elementi presenti in questa versione del coefficiente di correlazione di Pearson sono:

- $U$ : rappresenta l'insieme degli utenti che hanno valutato sia l'item  $i$  che l'item  $j$ .
- $r_{u,i}$ : è la valutazione dell'utente  $u$  relativo all'item  $i$ .
- $\bar{r}_i$ : è la media delle valutazioni degli utenti rispetto all'item  $i$ .
- $\bar{r}_j$ : è la media delle valutazioni degli utenti rispetto all'item  $j$ .

La predizione della valutazione sull'item  $i$  da parte dell'utente attivo si ottiene eseguendo una media ponderata:

$$p_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|}$$

Formula per il calcolo delle predizioni per l'approccio collaborativo *item to item*

### 3.2.2. Sistemi model based

I sistemi basati sul modello forniscono le raccomandazioni in base alle statistiche sulle preferenze degli utenti. Un prima approccio<sup>viii</sup> ai sistemi collaborativi suggerisce di trattare il problema come se fosse un problema di classificazione; quindi, viene costruito un classificatore per ogni utente attivo rappresentando gli elementi (*items*) come vettori di features<sup>1</sup> e le valutazioni dell'utente come le labels<sup>2</sup>.

L'approccio che viene maggiormente utilizzato fa uso dei *fattori latenti e della fattorizzazione di matrici*. Le tecniche di questo approccio si basano su nozioni statistiche che calcolano la similarità fra gli utenti, o fra gli elementi. Il modello a fattori latenti presume che la similarità tra utenti ed items sia indotta da fattori nascosti presenti nella struttura dati. *Es. un utente che valuta un film in modo positivo, probabilmente valuterà in modo positivo film che appartengono allo stesso genere o dello stesso regista.*

Nella fattorizzazione di matrici sono presenti un'ampia serie di tecniche utilizzate con successo che prevedono la rappresentazione simultanea degli utenti e degli item come un insieme di vettori di features sconosciute (*vettori colonna*)  $w_u$  per gli utenti,  $h_i$  per gli oggetti, lungo  $k$  dimensioni latenti. Queste vettori approssimano la valutazione dell'utente  $u$  relativa all'elemento  $i$  attraverso il prodotto  $w_u^T h_i$ , approssimando il rating  $r_{u,i}$  noto

<sup>1</sup> Feature: le caratteristiche che identificano un elemento (*item*)

<sup>2</sup> Label: l'etichetta valore su cui eseguire la predizione

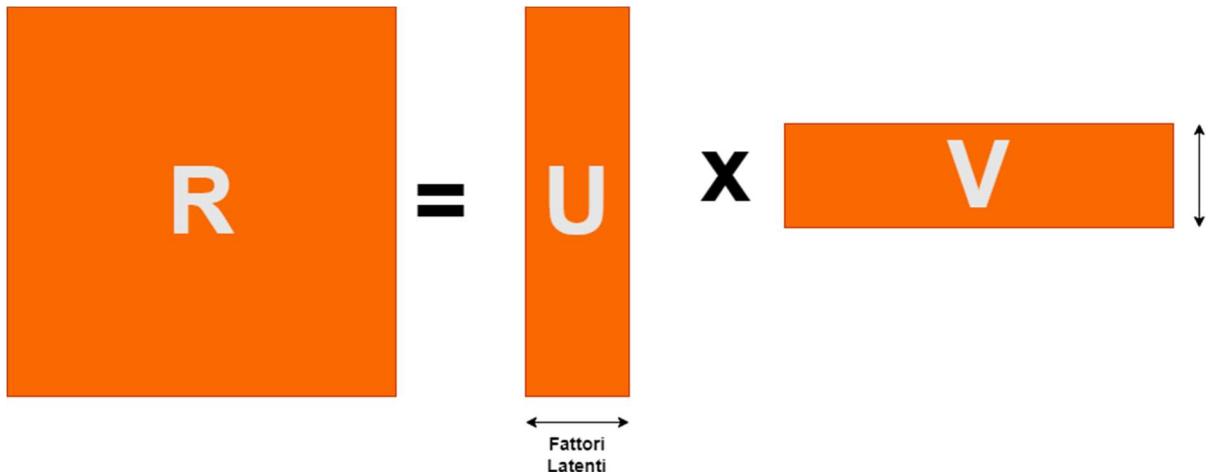
rispetto ad una funzione che misura la perdita d'informazione, spesso la funzione utilizzata è la perdita quadratica (*Squared loss*).

$$J(W, H, \{b_u\}_{u=1}^n, \{b_i\}_{i=1}^m) = \sum_{(u,i) \in L} (r_{u,i} - w_u^T h_i)^2$$

Formula funzione perdita quadratica

dove  $W$  e  $H$  sono matrici che rappresentano utenti e oggetti rispettivamente, e  $L$  è l'insieme di coppie utente-oggetto per cui le valutazioni sono conosciute.

Nei sistemi reali la matrice  $R$   $n * m$  contenente le valutazioni degli utenti (*users*) sugli elementi (*items*) sarà una matrice sparsa di grandi dimensioni, quindi il problema può essere ridotto suddividendo la matrice  $R$  nel prodotto di matrici dense di dimensione più piccola.



Fattorizzazione di matrici

Una tra le tecniche più usata nei sistemi reali è la fattorizzazione di matrici a valori singolari (*SVD*).

$$R = U \Sigma V^T$$

Fattorizzazione di matrici a valori singolati

Questo metodo permette di rappresentare  $R$  come il prodotto delle matrici  $U$ ,  $\Sigma$ ,  $V^T$  (*la trasposta di V*),  $R = U \Sigma V^T$ .  $U$  è una matrice ortogonale di rango  $m * m$ ,  $V^T$  è una matrice

ortogonale  $n * n$ ,  $\Sigma$  è una matrice diagonale a valori crescenti  $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p$ ,  $p = \min\{m, n\}$ , conosciuta come valori singolari di R.

### 3.3. Sistemi ibridi

I sistemi ibridi utilizzano tecniche che appartengono sia ai sistemi collaborativi che ai sistemi basati sul contenuto e calcolano le raccomandazioni usando entrambi gli approcci. Le implementazioni dei sistemi ibridi prevedono il calcolo delle predizioni sia con l'approccio basato sul contenuto che utilizzando l'approccio collaborativo separatamente e poi successivamente vengono combinate le previsioni ottenute da entrambi gli approcci, attraverso l'utilizzo di medie pesate.

### 3.4. Valutazione di un sistema

Un modello di apprendimento automatico viene valutato attraverso l'utilizzo di metriche che confrontano i valori predetti per il set di test con i valori reali. La misura più diffusa è il MAE (*Mean Absolute Error*)<sup>ix</sup>. Essa è definita come la sommatoria della differenza del valore assoluto tra il valore predetto e la valutazione media per l'utente  $u$  relativo all'item  $i$ .

$$MAE = \frac{\sum_{\{u,i\}} |p_{u,i} - r_{u,i}|}{N}$$

Formula del MAE – *Mean Absolute Error*

Una metrica correlata molto utilizzata è la RMSE (*Root Mean Squared Error*), la quale pone maggiore enfasi sul maggiore errore assoluto.

$$RMSE = \sqrt{\frac{\sum_{\{u,i\}} (p_{u,i} - r_{u,i})^2}{N}}$$

Formula RMSE – *Root Mean Squared Error*

### 3.5. Considerazioni in ambito assicurativo

Prima di descrivere le implementazioni effettuate per realizzare il sistema di raccomandazione è necessario puntualizzare alcuni aspetti relativi al dominio preso in esame, *l'ambito assicurativo*, per il quale è necessario utilizzare degli accorgimenti.

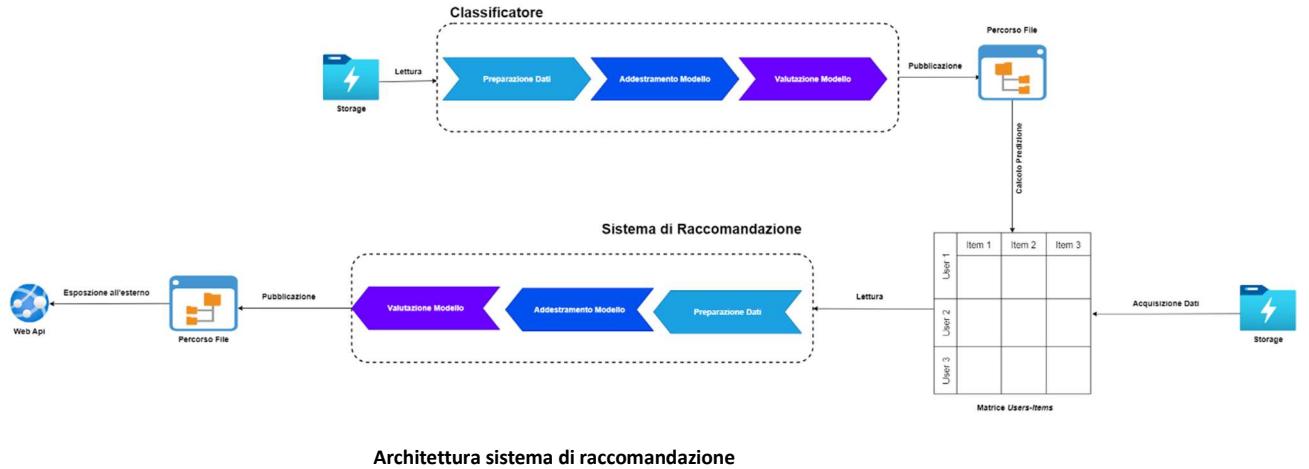
Le peculiarità dell'ambito assicurativo che possono essere individuate sono:

- **Cardinalità ridotta:** l'offerta assicurativa può contenere un numero limitato di prodotti rispetto all'offerta delle piattaforme e-commerce o delle piattaforme di streaming on-line. Questo è un aspetto che non può essere risolto in quanto è una caratteristica intrinseca del dominio applicativo.
- **Affidabilità:** un libro, la visione di un film sono prodotti di uso comune che vengono acquistati giornalmente o settimanalmente. Le coperture assicurative vengono modificate raramente pertanto è necessario avere un elevato livello di affidabilità da parte del cliente. Per affrontare questo aspetto è necessario implementare un sistema che abbia un'elevata affidabilità garantendo all'utente l'effettiva possibilità di proporre al cliente elementi che possano essere realmente di suo interesse.
- **Vincoli:** quando si sottoscrive un'assicurazione ci sono diversi vincoli che vanno rispettati. Per risolvere questo aspetto è indispensabile rappresentare i vincoli esistenti nel mondo assicurativo creando una base di conoscenza; alcuni degli strumenti attraverso cui è possibile rappresentare un grafo sono RDF<sup>x</sup> - *Resource Description Framework*, OWL<sup>xi</sup> - *Web Ontology Language* utilizzato per la rappresentazione di ontologie<sup>xii</sup>, Neo4j<sup>xiii</sup> per la rappresentazione di basi di dati a grafo.
- **Assenza di un rating esplicito:** nell'ambito assicurativo non è possibile chiedere un valore di gradimento esplicito al cliente e risulta difficile calcolarlo in modo implicito. Un approccio per la risoluzione di questa problematica può essere rappresentato dall'utilizzo di un metodo semi-supervisionato. In questo caso il rating viene calcolato da un algoritmo di apprendimento automatico.

Nel progetto presentato i problemi affrontati sono l'affidabilità e l'assenza di rating esplicito.

# 4. Soluzione implementata

L'immagine mostra l'architettura implementata per il sistema di raccomandazione realizzato.



Il sistema di raccomandazione è stato implementato seguendo un metodo semi-supervisionato. Il termine semi-supervisionato riguarda le valutazioni relative ai prodotti assicurativi in quanto non sono state acquisite dal cliente, il quale esprimendo il proprio criterio di gradimento in merito al servizio offerto oppure attraverso la compilazione di un questionario di gradimento; ma sono state definite come una funzione di probabilità calcolata da un algoritmo di apprendimento automatico, in seguito è stato addestrato il modello di raccomandazione sulle valutazioni calcolate in precedenza. Per calcolare le valutazioni è stato addestrato un algoritmo di classificazione, il quale per ogni cliente calcola la prossima polizza assicurativa che il cliente potrebbe sottoscrivere lo score ottenuto verrà impostato come valutazione nella matrice *users-items*. I passi definiti per la definizione del processo sono:

- Acquisizione dati.
- Analisi dei dati.
- Addestramento del modello di classificazione.
- Valutazione del classificatore.
- Popolamento della matrice *users-items*.
- Addestramento del modello di raccomandazione.
- Valutazione del sistema di raccomandazione.
- Rilascio modello.

## 4.1. Acquisizione dati

Il primo passo consiste nell'acquisizione e nell'analisi dei dati su cui eseguire l'addestramento del modello di classificazione per popolare la matrice *users-items*. I dati che vengono acquisiti dallo storage (*file in formato csv*) sono:

Nome Campo	Tipologia
Sesso	Carattere
Data di Nascita	Data
Stato Civile	Stringa
Coniuge a Carico	Stringa ( <i>true o false</i> )
Numero Figli	Numerico
Numero Figli a Carico	Numerico
Professione	Stringa
Reddito (RAL)	Numerico
Tipo Impiego	Stringa ( <i>Lavoratore dipendente/Libero professionista</i> )
Provincia di Residenza	Stringa ( <i>Sigla provincia</i> )
Polizza Stipulata	Nome polizza

Dati Cliente presenti nello Storage

Su questi dati vengono eseguite le fasi di pulizia e trasformazione preliminare dei dati che daranno origine alle *features* utilizzate dal classificatore. La fase di trasformazione del dato avviene definendo per i campi coinvolti delle funzioni di trasformazione dato.

$$\forall x \ F(x) \xrightarrow{\text{Funzione di Trasformazione}} y$$

Definizione funzione di trasformazione dati

Le funzioni di trasformazione che sono state definite sono:

- **Estrazione anno e mese di nascita:** la funzione prende in input la data di nascita del cliente e ritorna in output il mese e l'anno di nascita del cliente. Dalla data di nascita è stato eliminato il giorno in quanto utilizzare la data di nascita completa causava un'eccessiva variabilità sul dataset, non permettendo al modello di convergere e di conseguenza causare un'underfitting.
- **Estensione stato civile:** la funzione prende in esame lo stato civile del cliente e ritorna lo stato civile ed un valore booleano che identifica se il cliente è single oppure in coppia. La definizione del flag “*is-single*” è data dalla tabella:

Stato Civile	Is Single
Celibe/Nubile	True
Separato	True
Divorziato	True
Coniugato	False
Convivente	False

Tabella di conversione stato civile

- **Conversione coniuge a carico:** conversione del valore presente per coniuge a carico da stringa a booleano, se è presente un valore *null* o vuoto viene impostato *false*.
- **Conversione numero figli:** converte il valore presente in numerico, se è presente un valore *null* o vuoto viene impostato a 0.

- **Validazione numero figli a carico:** converte il valore presente in numerico, verifica che il valore calcolato sia minore o uguale del valore calcolato, se è presente un valore *null* o *vuoto* o fallisce la validazione viene impostato a 0.
- **Conversione tipo impiego:** viene definito un nuovo campo booleano *IsFreelancer* il quale viene valorizzato con true se il campo *Tipo Impiego* vale “*Libero professionista*” con false se vale “*Lavoratore dipendente*”.
- **Definizione tipo reddito:** viene definito un nuovo campo *Tipo Reddito* che definisce il tipo reddito del cliente ottenuto in base alla professione ed al valore del campo tipo impiego. Nella tabella sono mostrati i valori previsti per questa tipologia di campo ed i criteri applicati per la conversione:

Descrizione tipo reddito	Criterio applicato
Reddito da Lavoro Dipendente	Se Tipo Impiego vale <i>Lavoratore dipendente</i>
Reddito da Lavoro come Libero Professionista	Se Tipo Impiego vale <i>Libero professionista</i>
Nessun Reddito	Se professione corrisponde ad uno dei seguenti valori: <i>Casalinga, Studente, Disoccupato</i>
Altro Reddito	Se professione corrisponde ad uno dei seguenti valori: <i>Pensionato, Bracciante, Religioso, Società, Ente Privato, Ente Pubblico</i>

Tabella per la determinazione del tipo reddito

- **Estensione residenza:** partendo dalla sigla della provincia viene aggiunto un nuovo campo contenente il nome della regione a cui appartiene la provincia in cui risiede il cliente.
- **Estensione delle polizze sottoscritte:** partendo dal nome della polizza sottoscritta dal cliente vengono aggiunti i campi:
  - ✓ Identificativo polizza – è un numero progressivo che identifica univocamente il tipo polizza.
  - ✓ Codice polizza – è una stringa alfanumerica che identifica univocamente il tipo polizza.

Dopo aver eseguito le trasformazioni descritte le *features* del cliente saranno:

Nome Campo	Tipologia Campo
Mese Nascita	Numerico
Anno Nascita	Numerico
Stato Civile	Stringa
Is Single	Booleano
Coniuge Carico	Booleano
Numero Figli	Numerico
Numero Figli Carico	Numerico
Professione	Stringa
Reddito	Numerico
Tipo Reddito	Stringa
Is Freelancer	Booleano
Provincia Residenza	Stringa
Regione Residenza	Stringa
Nome Polizza	Stringa
Identificativo Polizza	Numerico
Codice Polizza	Stringa

Campi utilizzati per l'addestramento del classificatore

Le features definite nella tabella verranno utilizzate per estrarre il *customerId* durante il calcolo delle raccomandazioni per un nuovo cliente. Per eseguire l'addestramento del classificatore sono state rimosse le features:

- **Is Single:** la maggior parte dei clienti acquisiti durante i test hanno come stato civile (*celibe/Nubile, separato, divorziato*). Quindi la maggior parte delle volte questo campo è valorizzato con *False* e questo può causare overfitting<sup>xiv</sup> e quindi una degradazione delle prestazioni.
- **Coniuge Carico:** è un campo che dipende dallo stato civile, anche in questo caso la maggior parte delle volte questo campo è valorizzato con *False*, quindi un suo utilizzo può causare overfitting.
- **Numero figli a carico:** la maggior parte delle volte questo campo è valorizzato con *0*, la distribuzione degli altri valori non rappresenta un'informazione utile per potere eseguire l'addestramento.
- **Tipo reddito:** rappresenta una generalizzazione e di conseguenza una perdita d'informazione rispetto all'indicazione della professione.
- **Is Freelancer:** rappresenta una generalizzazione e di conseguenza una perdita d'informazione rispetto all'indicazione della professione, inoltre in presenza del campo *tipo impiego* risulta essere un'informazione ridondante.

I campi indicati verranno utilizzati nel momento in cui si acquisisce un nuovo cliente per la ricerca di clienti avente caratteristiche simili, in modo da risolvere il *cold start problem*<sup>xv</sup>.

## 4.2. Analisi dei dati

Sono stati acquisiti dal sistema 641.435 dati relativi a clienti che hanno sottoscritto le polizze. Prima di procedere ad eseguire l'addestramento del modello è necessario capire la distribuzione dei dati che viene mostrata dalla tabella e dal grafico che seguono:

Nome Polizza	Numero Clienti	Percentuale
R.C.A.	573.876	89,467522
INFORTUNI	64.266	10,0190978
MULTIGARANZIA ABITAZIONE	2.413	0,37618777
MALATTIA	766	0,11941974
R.C. DIVERSI	78	0,01216023
A.R.D.	22	0,00342981
GLOBALE FABBRICATI	8	0,0012472
INCENDIO/FURTO	4	0,0006236
TUTELA GIUDIZIARIA	2	0,0003118
<b>Totali</b>	<b>641.435</b>	<b>100</b>

Tabella con Distribuzione dei dati

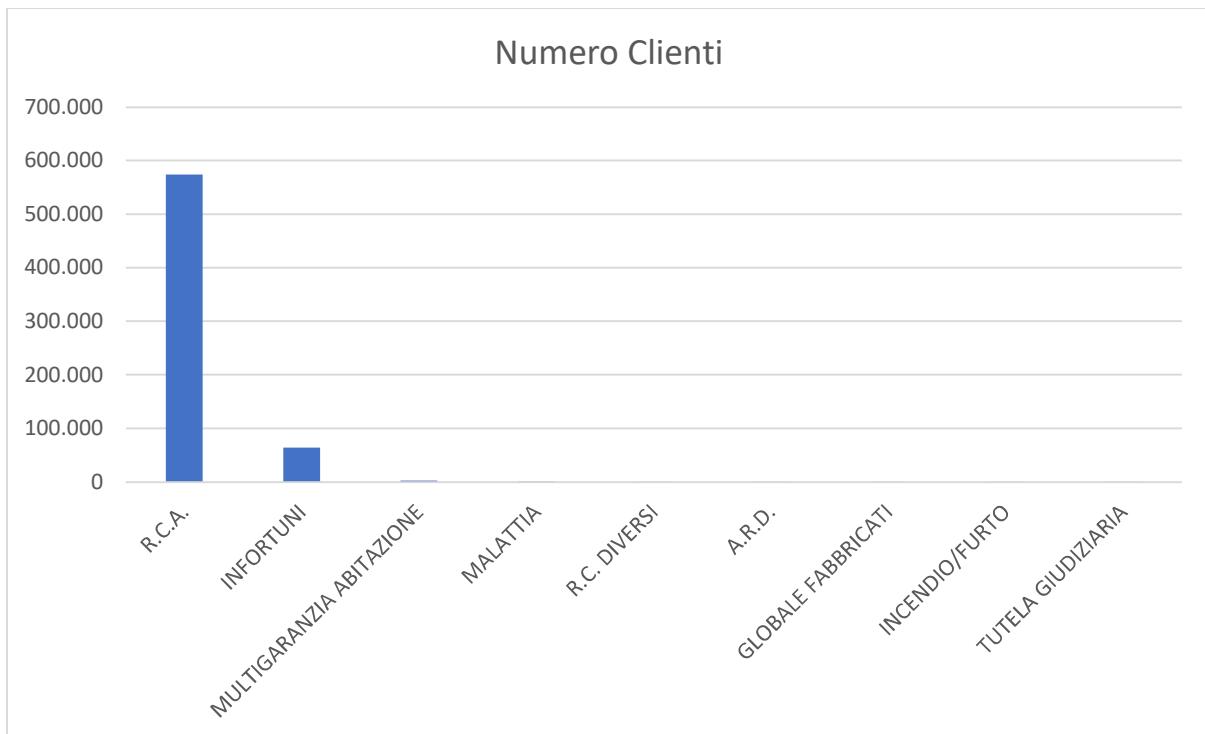


Grafico che mostra la distribuzione delle polizze acquistate dai clienti

Dai report mostrati è possibile notare che la distribuzione dei dati è fortemente sbilanciata verso l'R.C.A., l'89 % delle polizze stipulate sono R.C.A. il 10 % delle polizze stipulate sono infortuni il restante delle polizze sono valori infinitesimali, infatti, la loro somma è lo 0,51 %, questa è una situazione comunque reale in quanto all'interno di un nucleo familiare è possibile che ci siano più veicoli ed ognuno di essi deve avere l'assicurazione R.C.A. Inoltre, considerando il numero totale di clienti coinvolti e la distribuzione probabilmente ci sono clienti che hanno sottoscritto più di una polizza. I dati ottenuti vengono salvati su di una tabella del database (*CustomerLearningFeaturesTraining*). La tabella indicata viene utilizzata per eseguire l'addestramento del modello di classificazione. L'immagine mostra un'estrazione casuale di 15 record dalla tabella *CustomerLearningFeaturesTraining*.

Sesso	Mese Nascita	Anno Nascita	Stato Civile	Numero Figli	Professione	Reddito (R.A.L.)	Provincia Residenza	Regione Residenza	Identificativo Polizza	Codice Polizza
M	5	1965	Celibe/nubile	1	Disoccupato	0	TO	PIEMONTE	1	RCA012A1
M	1	1980	Divorziato	3	Dipendente generico	29795,303	NA	CAMPANIA	1	RCA012A1
F	8	1971	Convivente	2	Appartenente a Forze Armate	38789,26	LI	TOSCANA	1	RCA012A1
F	9	1973	Celibe/nubile	3	Operaio	25820,81	PG	UMBRIA	1	RCA012A1
M	5	1962	Vedovo	3	Operaio	19116,436	FI	TOSCANA	1	RCA012A1
M	8	1978	Convivente	3	Altra professione	18255,994	SA	CAMPANIA	1	RCA012A1
M	10	1980	Divorziato	2	Dipendente generico	31426,71	RM	LAZIO	1	RCA012A1
M	6	1987	Convivente	0	Dipendente generico	38403,496	MO	EMILIA ROMAGNA	1	RCA012A1
M	2	1979	Divorziato	2	Appartenente a Forze Armate	26219,613	RM	LAZIO	1	RCA012A1
F	8	1967	Convivente	0	Dipendente generico	22759,008	RM	LAZIO	1	RCA012A1
M	11	1952	Convivente	0	Pensionato	0	MI	LOMBARDIA	1	RCA012A1
F	7	1971	Divorziato	3	Insegnante	36984,12	MC	MARCHE	1	RCA012A1
F	5	1945	Divorziato	2	Casalinga	0	NA	CAMPANIA	1	RCA012A1
M	8	1943	Convivente	2	Artigiano	22623,29	NA	CAMPANIA	1	RCA012A1
M	2	1965	Vedovo	3	Insegnante	20049,953	NA	CAMPANIA	1	RCA012A1

Esempio di dati salvati sulla tabella *CustomerLearningFeatures*

I clienti vengono raggruppati per caratteristiche simili escludendo i dati delle polizze sottoscritte, i campi in base a cui viene eseguito il raggruppamento sono:

Nome Campo	Tipologia Campo
Sesso	Carattere
Mese Nascita	Numerico
Anno Nascita	Numerico
Stato Civile	Stringa
Is Single	Booleano
Coniuge Carico	Booleano
Numero Figli	Numerico
Numero Figli Carico	Numerico
Professione	Stringa
Is Freelancer	Booleano
Tipo Impiego	Stringa
Provincia Residenza	Stringa
Regione Residenza	Stringa

Tabella di raggruppamento clienti

Ai campi presenti in tabella viene aggiunto un ulteriore campo calcolato che verrà chiamato *CustomerId* esso è un contatore (*campo numerico*) che viene incrementato di un'unità per ogni occorrenza del cliente individuata, una volta terminato il calcolo del *CustomerId* per tutti i clienti vengono aggiornati i dati. Il calcolo del *CustomerId* è un calcolo che avviene in modo incrementale. Di seguito viene mostrato lo pseudo-codice per il calcolo descritto. Pseudo-codice per il calcolo del *CustomerId*:

```

Procedure UpdateCustomerFeaturesWithCustomerId()
    customerFeatures = GetCustomerFeatures()
    lastCustomerId = GetMaxCustomerId()
    For item in customerFeatures
        item.CustomerId = lastCustomerId
        lastCustomerId = lastCustomerId + 1
    Endfor
    UpdateCustomerFeatures(customerFeatures)
End

```

Pseudo-codice per il calcolo del *CustomerId*

Una volta calcolato il valore di *CustomerId* per tutti i clienti acquisiti i dati vengono salvati all'interno di una tabella del database (*CustomerLearningFeatures*) in modo da potere essere ricercati in seguito. L'immagine mostra un'estrazione casuale di 15 clienti d'esempio dove viene mostrato l'esito dell'applicazione dell'algoritmo per il calcolo del *CustomerId*.

Customer Id	Sesso	Mese Nascita	Anno Nascita	Stato Civile	Is Single	Coniuge Carico	Numero Figli	Numero Figli Carico	Professione	Is Freelancer	Reddito (R.A.L.)	Tipo Reddito	Provincia Residenza	Regione Residenza	Identificativo Polizza	Codice Polizza
93010F		3	1975	Vedovo	1	0	2	1	Liber professionista	1	49893,17	Liber professionista	CO	LOMBARDIA	1 RC012A1	
364102M		2	1978	Convivente	0	0	2	1	Altra professione	1	21327,77	Liber professionista	BO	EMILIA ROMAGNA	1 RC012A1	
264518M		10	1975	Vedovo	1	0	0	0	Autista	0	6240,47	Liber professionista	MI	LOMBARDIA	1 RC012A1	
99421F		3	1994	Convivente	0	0	1	0	Operario	0	9561,14	Lavoratore Dipendente	VA	LOMBARDIA	1 RC012A1	
242431M		1	2000	Divorzia-to	1	0	0	0	Operario	0	34521,39	Lavoratore Dipendente	CO	LOMBARDIA	1 RC012A1	
285411M		11	1961	Vedovo	1	0	0	0	Commerciale	1	27084,43	Liber professionista	PT	TOSCANA	1 INF0H1F6	
112128F		4	1979	Vedovo	1	0	1	0	Altra professione	1	35927,47	Liber professionista	PA	SICILIA	1 RC012A1	
654412M		7	2000	Convivente	0	0	0	0	Dipendente generico	0	10474,29	Lavoratore Dipendente	CL	SICILIA	1 RC012A1	
528936M		7	1966	Convivente	0	0	2	0	Dipendente generico	0	19433,96	Lavoratore Dipendente	NA	CAMPANIA	1 RC012A1	
388478M		3	1970	Convivente	0	1	1	0	Altra professione	1	23769,95	Liber professionista	PD	VENETO	1 INF0H1F6	
429439M		4	1974	Convivente	0	0	2	0	Imprenditore	1	45753,31	Liber professionista	FC	EMILIA ROMAGNA	1 RC012A1	
368369M		2	1983	Coniugato	0	0	0	0	Lavoratore autonomo	1	11042,34	Liber professionista	VR	VENETO	1 INF0H1F6	
278947M		11	1943	Convivente	0	0	0	0	Pensionato	0	0	Nessun Reddito	RM	LAZIO	1 RC012A1	
388232M		3	1969	Coniugato	0	1	2	0	Impiegato	0	27366,86	Liber professionista	SP	LIGURIA	1 INF0H1F6	
263639M		10	1974	Vedovo	1	0	1	0	Operario	0	24278,29	Lavoratore Dipendente	MI	LOMBARDIA	1 RC012A1	

Esempio di dati salvati sulla tabella *CustomerLearningFeatures*

### 4.3. Valutazione del modello

Per implementare gli algoritmi di apprendimento automatico è stata utilizzata ML.NET<sup>xvi</sup> una libreria di machine learning sviluppata da Microsoft, la quale permette di costruire un modello attraverso l'utilizzo di una GUI (*AutoML*), esegue il training utilizzando diversi algoritmi e poi suggerisce quello che presenta le prestazioni migliori. In ML.NET sono state

definite delle metriche per la valutazione dei modelli, i modelli di classificazione multi-classe vengono valutati attraverso le seguenti metriche:

- **Micro-accuracy:** L'accuratezza micro-media aggrega i contributi di tutte le classi per calcolare la metrica media. Corrisponde alla percentuale di istanze stimate correttamente. La micro-media non tiene conto dell'appartenenza a una classe. Essenzialmente, ogni coppia campione-classe contribuisce nello stesso modo alla metrica di accuratezza. L'obiettivo di questa media è quello di tendere a 1,0.
- **Macro-accuracy:** L'accuratezza macro-media corrisponde all'accuratezza media a livello di classe. Viene confrontata l'accuratezza per ogni classe e l'accuratezza macro-media è la media di queste accuratezze. Essenzialmente, ogni classe contribuisce nello stesso modo alla metrica di accuratezza. Alle classi di minoranza viene assegnato un peso uguale a quello delle classi più grandi. La metrica della macro-media assegna lo stesso peso a ogni classe, indipendentemente dal numero di istanze di tale classe contenute nel set di dati. L'obiettivo di questa metrica è quello di tendere a 1,0.
- **Log-loss:** La perdita logaritmica misura le prestazioni di un modello di classificazione in cui l'input della stima è un valore di probabilità compreso tra 0,00 e 1,00. Questa metrica aumenta quando la probabilità stimata devia dall'etichetta effettiva. L'obiettivo di questa metrica è quello di tendere a 0,0.
- **Log-loss reduction:** La riduzione della perdita logaritmica può essere interpretata come un vantaggio del classificatore rispetto alla stima casuale. Questa metrica può assumere valore compreso tra 1,00 e 0,00, i valori che tendono a 1,00 sono stime migliori rispetto a quelle che tendono a 0,00.

I modelli di raccomandazione utilizzano le stesse metriche utilizzate per i modelli di regressione visto che entrambi producono un valore numerico. I modelli di regressione producono un valore numerico che rappresenta l'output di una generica funzione, i modelli di raccomandazione producono in output un numero compreso tra un intervallo, valore *di classificazione*, di valori (es. *una valutazione da 1 a 5*) oppure produce in output una raccomandazione *si/no* oppure *0/1*. Le metriche utilizzate per valutare i modelli di regressione o di raccomandazione sono:

- **R-Squared:** R-squared ( $R^2$ )<sup>xvii</sup> o coefficiente di determinazione rappresenta la potenza predittiva del modello come valore compreso tra  $-\infty$  e 1,00. 1,00 significa corrispondenza perfetta e la corrispondenza può essere arbitrariamente insufficiente, quindi i punteggi possono essere negativi. Il punteggio 0,00 significa che il modello indovina il valore previsto per l'etichetta. Un valore  $R^2$  negativo indica che l'adattamento non segue la tendenza dei dati e il modello esegue prestazioni peggiori rispetto all'ipotesi casuale. Ciò è possibile solo con modelli di regressione non lineare o regressione lineare vincolata.  $R^2$  misura il grado di prossimità dei valori dei dati di test effettivi ai valori stimati. L'obiettivo di questa metrica è tendere a 1,0.
- **Absolute-loss:** Absolute-loss<sup>xviii</sup> o errore assoluto medio misura la prossimità delle stime ai risultati effettivi. Corrisponde alla media di tutti gli errori del modello, dove un errore del modello è la distanza tra il valore di etichetta stimato e quello corretto. Questo errore di stima viene calcolato per ogni record del set di dati di test. Infine,

viene calcolato il valore medio per tutti gli errori assoluti registrati. L'obiettivo di questa metrica è tendere a 0,0.

- **Squared-loss:** La perdita quadratica<sup>xix</sup> o l'errore quadratico medio (MSE), detto anche deviazione quadratica media (MSD), indica come chiudere una linea di regressione a un set di valori di dati di test prendendo le distanze dai punti alla linea di regressione (queste distanze sono gli errori E) ed elevandoli al quadrato. La quadratura assegna più peso alle differenze maggiori. Questa metrica assume sempre valori positivi, il suo obiettivo è tendere a 0,0.
- **RMS-loss:** RMS-loss<sup>xx</sup> o radice dell'errore quadratico medio, anche detto radice della deviazione quadratica media, misura la differenza tra i valori stimati da un modello e i valori osservati nell'ambiente del modello. RMS-loss è la radice quadrata di Squared-loss e ha la stessa unità come etichetta, simile a absolute-loss ma assegnando più peso alle differenze maggiori. La radice dell'errore quadratico medio viene comunemente usata in climatologia, previsioni e analisi di regressione per verificare i risultati sperimentali. Questa metrica misura l'accuratezza del modello, assume sempre valori positivi ed il suo obiettivo è tendere a 0,0.

## 4.4. Sistema di raccomandazione

In ML.NET i sistemi di raccomandazione sono stati implementati utilizzando l'approccio collaborativo ed il modello basato sulla fattorizzazione di matrici (*matrix factorization*)<sup>xxi</sup>, il primo passo da seguire per questo approccio è popolare la matrice *users-items*.

### 4.4.1. Approccio collaborativo

La matrice *users-items* è stata popolata inserendo per ogni *CustomerId* calcolato di un “1” in corrispondenza delle polizze sottoscritte dal cliente ed uno “0” altrimenti, l'algoritmo descritto viene mostrato dallo pseudo-codice che segue.

```

Function UpdateMatrixUserItems()
    lastUserId = GetMaxUserId()
    customerFeatures = GetCustomerFeatures(lastUserId)
    InsurancePolicyCategories = GetInsurancePolicyCategories()
    For itemCustomer in customerFeatures
        customerInsurancePolicies = getInsurancePolicies(itemCustomer.CustomerId)

        For itemPolicy in InsurancePolicyCategories

            if (customerInsurancePolicies.Contains(itemPolicy.Id))
                matrixUserItems[itemCustomer.CustomerId][itemPolicy.Id] = 1
            Else
                matrixUserItems[itemCustomer.CustomerId][itemPolicy.Id] = 0
            End

        Endfor

    Endfor
    return matrixUserItems
End

```

**Pseudo-codice per popolamento matrice *users-items***

L'algoritmo produce in output una matrice *users-items* sparsa e fortemente sbilanciata per le polizze R.C.A. come è possibile vedere dall'immagine.

utente	R.C.A.	A.R.D.	R.C. DIVERSI	MULTIGARANZIA ABITAZIONE	GLOBALE FABBRICATI	INFORTUNI	MALATTIA	INCENDIO/FURTO	TUTELA GIUDIZIARIA
483374	1	0	0	0	0	1	0	0	0
483080	1	0	0	0	0	0	1	0	0
482878	1	0	0	0	0	1	0	0	0
482875	1	0	0	0	0	1	0	0	0
482873	1	0	0	0	0	1	0	0	0
482830	1	0	0	0	0	1	0	0	0
482806	1	0	0	0	0	1	0	0	0
482671	1	0	0	0	0	1	0	0	0
482549	1	0	0	0	0	1	0	0	0
482266	1	0	0	0	0	1	0	0	0

Matrice *users-items*

Eseguendo l'addestramento del modello utilizzando la matrice *users-items* descritta, si ottiene un modello con prestazioni decisamente scadenti. Nel report vengono mostrate le cinque migliori iterazioni ottenute eseguendo l'addestramento con la matrice *users-items* creata.

Algoritmo	RSquared	Absolute-loss	Squared-loss	RMS-loss	Durata	Iterazione
MatrixFactorization	0,5076	0,08	0,05	0,22	0,5	186
MatrixFactorization	0,5075	0,08	0,05	0,22	0,5	203
MatrixFactorization	0,5073	0,08	0,05	0,22	0,5	199
MatrixFactorization	0,5069	0,08	0,05	0,22	0,6	205
MatrixFactorization	0,5069	0,08	0,05	0,22	0,5	202

Primi cinque modelli per il sistema di raccomandazione

Addestrare un algoritmo di machine learning richiede l'utilizzo di molte risorse, quindi è possibile concludere che un algoritmo con prestazioni intorno al 0,5 (50 %) è inutilizzabile,

per essere impiegato in scenari reali un algoritmo dovrebbe avere delle prestazioni minime del 0,90 (90 %) un algoritmo con prestazioni inferiori può essere utilizzato solo per scopi didattici, *il lancio di una moneta ha uno score del 0,5 (50 %) a costo zero.*

*Melville ed altri<sup>xxii</sup>* hanno proposto l'uso di una rete neurale bayesiana<sup>xxiii</sup> per incrementare le prestazioni di un algoritmo di raccomandazione, essi hanno utilizzato una BNNs<sup>3</sup> per calcolare la probabilità per le coppie *user-item* che hanno rating 0 in modo da ottenere una matrice users-items densa e come conseguenza le prestazioni del sistema di raccomandazione dovrebbero aumentare.

#### 4.4.2. Addestramento classificatore

Seguendo l'approccio proposto da Melville ed altri, ho implementato un classificatore, il quale prende in input le caratteristiche (*features*) definite per i clienti ed alcune caratteristiche della polizza sottoscritta (*identificativo, codice, nome e descrizione*) e predice la prossima polizza che il cliente potrebbe sottoscrivere, la percentuale di predizione (*score*) rappresenterà il rating che verrà salvato nella matrice users-items.

Per eseguire l'addestramento del modello sono stati seguiti i passi:

- Selezione la sorgente dati (*in questo caso la connessione al database sql server*).
- Selezionare la tabella o la vista contenente i dati su cui eseguire l'addestramento.
- Specificare le opzioni per le colonne che rappresentano le features
  - ✓ È possibile includere o ignorare la feature.
  - ✓ La feature può essere impostata come continua o categorica.
- Identificare la colonna che rappresenta la label da predire.

Dopo aver configurato il modello di classificazione è stato eseguito l'addestramento (*training*) e la valutazione del modello utilizzando diversi algoritmi. Dopo aver completato la procedura il sistema ha proposto il modello con le prestazioni migliori. Il report mostra i modelli addestrati e le loro prestazioni. Il modello suggerito è stato il *LbfgsMaximumEntropyMulti*, il quale è uno fra i cinque modelli con le prestazioni migliori.

Algoritmo	Micro Accuracy	Macro Accuracy	Durata	Iterazione
SdcaMaximumEntropyMulti	0,8935	0,2007	6,0	0
FastForestOva	0,9986	0,6000	9,1	1
SdcaLogisticRegressionOva	0,8931	0,2000	11,9	2
SdcaLogisticRegressionOva	0,9870	0,3837	17,1	3
LbfgsLogisticRegressionOva	0,9999	0,7929	5,9	4
FastTreeOva	0,9999	0,8000	6,8	5
LbfgsMaximumEntropyMulti	0,9999	0,8000	3,3	6
FastForestOva	1,0000	0,8333	66,0	7
LightGbmMulti	0,9997	0,7647	2,7	8
SdcaMaximumEntropyMulti	0,8931	0,1667	4,3	10
LbfgsLogisticRegressionOva	0,9991	0,5677	20,7	11
FastTreeOva	0,9999	0,8000	4,3	12
LbfgsMaximumEntropyMulti	0,9984	0,5846	2,0	13

---

<sup>3</sup> Bayesian Neural Networks

FastForestOva	1,0000	0,7143	96,4	14
LightGbmMulti	0,9986	0,6000	2,2	15
SdcaLogisticRegressionOva	0,8931	0,1429	117,6	16
LbfgsMaximumEntropyMulti	1,0000	0,8571	15,0	17
FastTreeOva	0,9999	0,6667	9,8	18
LbfgsLogisticRegressionOva	0,9999	0,8000	3,5	19
SdcaMaximumEntropyMulti	0,8939	0,2015	2,0	20
FastForestOva	0,9999	0,6667	22,4	21
FastForestOva	1,0000	0,7143	113,4	22

Tabella riassuntiva delle valutazioni dei modelli di classificazione

#### 4.4.3. Approccio ibrido

Terminato l'addestramento del classificatore si procede a popolare nuovamente la matrice users-items. In questo caso l'algoritmo per ogni customerId segue i passi:

- Se il cliente ha sottoscritto la polizza corrente viene assegnato alla coppia *customerId-identificativoPolizza* "1" come valore di rating.
- Altrimenti si procede a calcolare la predizione per il customerId che si sta elaborando su quella polizza, se il valore di predizione ottenuto corrisponde viene assegnato alla coppia *customerId-identificativoPolizza* lo *score* ottenuto come valore della predizione, *un valore compreso nell'intervallo [0..1]*.
- Altrimenti alla coppia *customerId-identificativoPolizza* viene assegnato "0" come valore di rating.

L'immagine che segue mostra lo pseudo-codice relativo all'algoritmo appena descritto.

```

Function updateMatrixUsersItems()
    matrixUsersItems = new [int][int]
    lastUserId = getMaxUserId()
    biasFactor = calculateBiasFactor(lastUserId)
    customerFeatures = getCustomerFeatures(lastUserId)
    insurancePolicyCategories = getInsurancePolicyCategories()

    For itemCustomer in customerFeatures

        customerInsurancePolicies = getInsurancePolicies(itemCustomer.CustomerId)

        For itemPolicy in insurancePolicyCategories

            If (customerInsurancePolicies.Contains(itemPolicy.Id))
                matrixUsersItems[itemCustomer.CustomerId] = 1
            Else
                prediction = calculatePrediction(itemCustomer.CustomerId, itemPolicy.Id)
                If (prediction.PredictLabel == itemPolicy.Id)
                    matrixUsersItems[itemCustomer.CustomerId][itemPolicy.Id] = prediction.Score
                Else
                    matrixUsersItems[itemCustomer.CustomerId][itemPolicy.Id] = 0

            EndIf

        EndFor

    EndFor

    return matrixUsersItems

End

```

#### Algoritmo di popolamento Matrice Users-Items con approccio ibrido

L'esecuzione dell'algoritmo indicato permette di ottenere una matrice meno sparsa, *presenza di coppie users-items con valore nullo (0)*, questo è un passaggio fondamentale nella definizione dell'algoritmo di raccomandazione in quanto inserendo la probabilità solo in corrispondenza delle polizze effettivamente sottoscritte dai clienti si otterrebbe come risultato una matrice sparsa e fortemente sbilanciata per le polizze R.C.A. quindi il sistema di raccomandazione fornirà come suggerimento le polizze R.C.A. con una probabilità molto vicina al 100 % le altre polizze non verrebbero considerate. L'immagine mostra un esempio della matrice *users-items* dopo aver eseguito l'algoritmo.

Utente	R.C.A.	A.R.D.	R.C. DIVERSI	MULTIGARANZIA ABITAZIONE	GLOBALE FABBRICATI	INFORTUNI	MALATTIA	INCENDIO/FURTO	TUTELA GIUDIZIARIA
608156	1	0,312825382	0,992839336	0,998143971	0,546862423	0,999956131	0,998504817	0	0
608155	1	0,561366558	0,984298348	0,997675717	0,525521338	0,999937057	0,997343719	0,292187572	0
608154	1	0,421329409	0,98940587	0,998552442	0,690226853	0,999954224	0,997448385	0	0
608110	1	0,631929517	0,985838532	0,997996449	0,540777147	0,999928474	0,996552706	0,249064803	0
608102	1	0,293249071	0,990594566	0,999236405	0,730697334	0,999944687	0,997449338	0	0
608101	1	0,553846896	0,986748576	0,997986913	0,565839708	0,999926567	0,996529877	0,292550504	0
608029	1	0	0,987695754	0,999288797	0,48910445	0,999970436	0,997610092	0	0
607588	1	0,649638295	0,926172316	0,997201085	0	0,999913216	0,996139348	0	0
607587	1	0,542169333	0,956773359	0,997343719	0	0,999914169	0,996342659	0	0
607586	1	0,732957184	0,926918089	0,997393191	0	0,999920845	0,996146023	0	0
607556	1	0,37024498	0,963673651	0,997585356	0,242373884	0,999920845	0,996473849	0	0,419717133
484114	0,999988556	0,326313287	0,983309448	0,997351348	0,392030209	1	0,996880651	0	0,336352408
484113	0,999974251	0,477903903	0,982899725	0,997548282	0,452503979	1	0,995702446	0	0,470483571

Matrice users-items

La matrice users-items iniziale era una matrice sparsa con una presenza di celle valorizzate con "0" per l'88,9 %, come viene mostrato nella tabella e nel grafico che seguono.

Nome Polizza	Clienti con celle con valore 1 per polizza	Clienti con celle con valore 0 per polizza
R.C.A.	541.776	66.412
A.R.D.	22	608.166
R.C. DIVERSI	78	608.110
MULTIGARANZIA ABITAZIONE	2.394	605.794
GLOBALE FABBRICATI	8	608.180
INFORTUNI	63.619	544.569
MALATTIA	764	607.424
INCENDIO/FURTO	4	608.184
TUTELA GIUDIZIARIA	2	608.186

Distribuzione iniziale matrice users-items

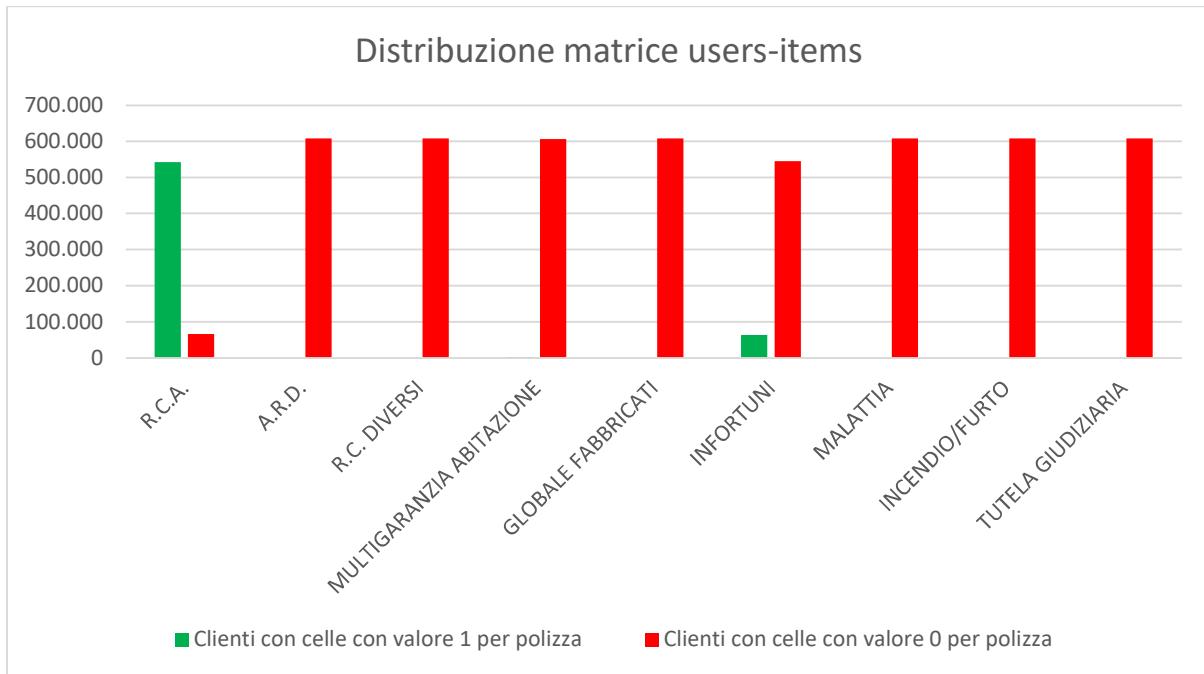


Grafico distribuzione matrice users-items

L'applicazione dell'algoritmo di classificazione riduce i valori nulli presenti nella matrice users-item al 25,1 %, è una percentuale alta. Addestrando l'algoritmo di raccomandazione con questa configurazione della matrice users-items l'algoritmo va in overfitting in quanto la valutazione migliore che riesce a raggiungere è dell'0,85 (85 %) per un dataset con al più 12.000 (*per un totale di 108.000 elementi*) utenti, provando ad aumentare il numero degli utenti le prestazioni dell'algoritmo diminuiscono fino a raggiungere un valore inferiore all'0,50 (50 %).

Il 50 % era la valutazione rilevata durante l'addestramento del primo algoritmo di raccomandazione, *il caso in cui la matrice users-items era composta solo da 0 e 1.*

Per risolvere il problema relativo all'overfitting del sistema di raccomandazione ho seguito i passi descritti di seguito:

- Rimozione dal dataset utilizzato per addestrare il classificatore le features ripetitive.
- Sono state rimosse le features:
  - ✓ Is single.
  - ✓ Numero figli a carico.
  - ✓ Fascia reddito.
  - ✓ Tipo reddito.
  - ✓ Is freelancer.
- Addestramento del modello di classificazione utilizzando le features considerate definitive.
- Popolamento della matrice users-items utilizzando il nuovo classificatore.
- Addestramento del modello di raccomandazione.

La matrice users-items ottenuta presenta valori nulli 0, solo per le coppie:

- Utente-INCENDIO/FURTO
- Utente-TUTELA GIURIDICA

Le polizze *incendio/furto* e *tutela giuridica* sono le polizze sottoscritte con minore frequenza all'interno del dataset preso in esame, 4 e 2 *occorrenze rispettivamente*.

La valutazione del modello è 0,93 (93 %). L'immagine mostra un'estrazione causale di 15 utenti presenti nella matrice users-item popolata.

Utente	R.C.A.	A.R.D.	R.C. DIVERSI	MULTIGARANZIA ABITAZIONE	GLOBALE FABBRICATI	INFORTUNI	MALATTIA	INCENDIO/FURTO	TUTELA GIUDIZIARIA
30015	1	0,996179	0,9981432	0,9983018	0,67827082	0,99978966	0,99836087	0	0
21854	1	0,9962	0,9981	0,9983	0,8304	0,9998	0,9983	0	0
4504	1	0,9962	0,9981	0,9983	0,6333	0,9998	0,9984	0	0
18689	0,9999	0,9961	0,9981	0,9983	0,8293	1	0,9983	0	0
19140	1	0,9961	0,9981	0,9983	0,8479	0,9998	0,9983	0	0
38911	1	0,996182	0,99814498	0,99830341	0,63334656	0,9997912	0,99836248	0	0
16441	1	0,9962	0,9981	0,9983	0,6333	0,9998	0,9984	0	0
38815	1	0,996182	0,99814498	0,99830341	0,63334656	0,9997912	0,99836248	0	0
26953	1	0,996179	0,9981432	0,9983018	0,67827082	0,99978966	0,99836087	0	0
34787	1	0,996182	0,99814498	0,99830341	0,63334656	0,9997912	0,99836248	0	0
33531	1	0,996182	0,99814498	0,99830341	0,63334656	0,9997912	0,99836248	0	0
18440	1	0,9962	0,9981	0,9983	0,6333	0,9998	0,9984	0	0
28903	1	0,996179	0,9981432	0,9983018	0,67827082	0,99978966	0,99836087	0	0
27871	1	0,996179	0,9981432	0,9983018	0,67827082	0,99978966	0,99836087	0	0
2575	1	0,9962	0,9981	0,9983	0,6333	0,9998	0,9984	0	0

Matrice users-items popolata utilizzando il classificatore

Dall'esempio mostrato nell'immagine si possono notare dei particolari:

- I valori presenti nella matrice users-items sono compresi tra 0 e 1.
- Gli elementi relativi alle polizze *incendio/furto* e *tutela giuridica* hanno valore 0.

- Gli elementi relativi alle altre polizze hanno tutti valori simili tra loro.

Queste caratteristiche presenti nella matrice users-item all'aumentare degli utenti potrebbero portare l'algoritmo di raccomandazione in overfitting.

Per risolvere questi problemi ho eseguito gli interventi:

- Ho portato i valori di rating su di una scala più ampia, *scala percentuale*.
- Ho aggiunto un errore, *bias*, alle valutazioni, *valore causale compreso tra 0 e 1*.

Nell'immagine che segue viene mostrato l'algoritmo per il popolamento della matrice users-items utilizzando le predizioni del classificatore e applicando il bias.

```
Function updateMatrixUsersItems()
    matrixUsersItems = new [int][int]
    lastUserId = getMaxUserId()
    biasFactor = calculateBiasFactor(lastUserId)
    customerFeatures = getCustomerFeatures(lastUserId)
    insurancePolicyCategories = getInsurancePolicyCategories()

    For itemCustomer in customerFeatures
        customerInsurancePolicies = getInsurancePolicies(itemCustomer.CustomerId)

        For itemPolicy in insurancePolicyCategories
            If (customerInsurancePolicies.Contains(itemPolicy.Id))
                matrixUsersItems[itemCustomer.CustomerId] = (1 * 100) + biasRandom()
            Else
                prediction = calculatePrediction(itemCustomer.CustomerId, itemPolicy.Id)
                If (prediction.PredictLabel == itemPolicy.Id)
                    matrixUsersItems[itemCustomer.CustomerId][itemPolicy.Id] = (prediction.Score * 100) + biasRandom()
                Else
                    matrixUsersItems[itemCustomer.CustomerId][itemPolicy.Id] = biasRandom()

            EndFor
        EndFor

    return matrixUsersItems
End
```

Algoritmo di popolamento della matrice users-items con approccio ibrido ed applicazione bias

Il risultato di questi interventi è una matrice densa, *senza valori nulli*. Nell'immagine che segue viene mostrato l'algoritmo di popolamento della matrice users-items utilizzando gli accorgimenti descritti.

Utente	R.C.A.	A.R.D.	R.C. DIVERSI	MULTIGARANZIA ABITAZIONE	GLOBALE FABBRICATI	INFORTUNI	MALATTIA	INCENDIO/FURTO	TUTELA GIUDIZIARIA
88035	100,8702931	100,3642838	100,672434	100,5224459	63,59157555	100,4997035	100,5588875	0,28185015	0,10514831
117624	100,66388232	99,80402476	99,92864311	100,5107902	83,35666493	100,159472	100,0859721	0,42021056	0,29157285
40124	100,4367197	100,080489	99,99032022	100,2917925	85,1834103	100,0706081	100,5459405	0,05848456	0,14064841
87870	100,0280726	99,7896379	99,97248491	99,95307594	63,59249063	100,7305868	100,7799046	0,42190716	0,89005775
41639	100,5350826	100,2559563	99,99793876	100,4054374	63,6415394	100,4119402	99,95525767	0,50482208	0,13362255
141150	100,6650618	99,67575869	100,5119298	100,5208042	84,93771555	100,9426164	100,7457855	0,48312247	0,83470794
79019	100,2828986	100,348417	100,508473	100,6889284	63,59550667	100,1482053	100,3134065	0,75739538	0,37290636
8546	100,5351345	100,1862105	100,4074212	100,6660111	68,16233892	100,1122929	100,7076117	0,1754101	0,03801789
124934	100,3151793	100,393429	100,1797103	99,99029652	67,94334547	100,3301379	99,89845836	0,67566415	0,702233
169407	100,7384532	100,0858363	100,7827764	99,86347532	63,43548941	100,6313055	100,0954791	0,05003606	0,74610892
70630	100,1404877	100,2818456	99,98171063	100,286242	63,39240973	100,5114148	100,0140488	0,45576473	0,28759732
141516	100,4005079	100,1339136	100,0807713	100,466604	64,25903337	100,7805662	100,1731476	0,44947801	0,14207934
137039	100,4667027	100,2997074	100,0676593	100,3496348	63,44124515	100,0453203	100,0131902	0,91443703	0,02818241
206993	100,1864824	99,9375129	100,3110119	100,6105623	83,88199948	100,1561758	100,667627	0,903815808	0,466906394
20248	100,6994825	100,5925165	100,3617175	100,1065605	63,36899587	100,3380415	100,3913577	0,2908911	0,70390735

Matrice users-items popolata utilizzando l'approccio ibrido e l'applicazione del bias

La valutazione del sistema di raccomandazione su tutti i clienti acquisiti è dello 0,9778 (97,78 %) come indicato dal report, sono indicati i 5 migliori risultati ottenuti dall'addestramento del modello.

Algoritmo	RSquared	Absolute Loss	Squared Loss	RMS Loss	Durata	Iterazione
MatrixFactorization	0,9778	2,90	37,15	6,10	42,5	12
MatrixFactorization	0,9726	3,08	45,78	6,77	180,9	13
MatrixFactorization	0,8652	7,45	224,32	14,98	9,3	2
MatrixFactorization	0,3692	20,48	1043,02	32,30	1,3	11
MatrixFactorization	0,0000	0,00	0,00	0,00	76,6	10

Primi cinque modelli per il sistema di raccomandazione

La matrice users-items costruita per addestrare il modello di raccomandazione contiene elementi fra loro simili.

- La prima matrice costruita utilizzando le polizze sottoscritte dal cliente, contiene valori 0 e 1; il modello di raccomandazione ha raggiunto una valutazione dello 0,5076.
- La seconda matrice costruita utilizzando la prima versione del classificatore, contiene valori 0, 1 e la probabilità valore compreso tra 0 e 1; il modello di raccomandazione ha raggiunto una valutazione dello 0,8562.
- La terza matrice costruita utilizzando la seconda versione del classificatore, contiene valori 0, 1 e la probabilità valore compreso tra 0 e 1; il modello di raccomandazione ha raggiunto una valutazione dello 0,9375.
- La quarta matrice costruita utilizzando la seconda versione del classificatore, applicando una conversione in scala percentuale ed un bias, contiene valori compresi tra  $0 + bias$  e  $100 + bias$ ; il modello di raccomandazione ha raggiunto una valutazione dello 0,9951.

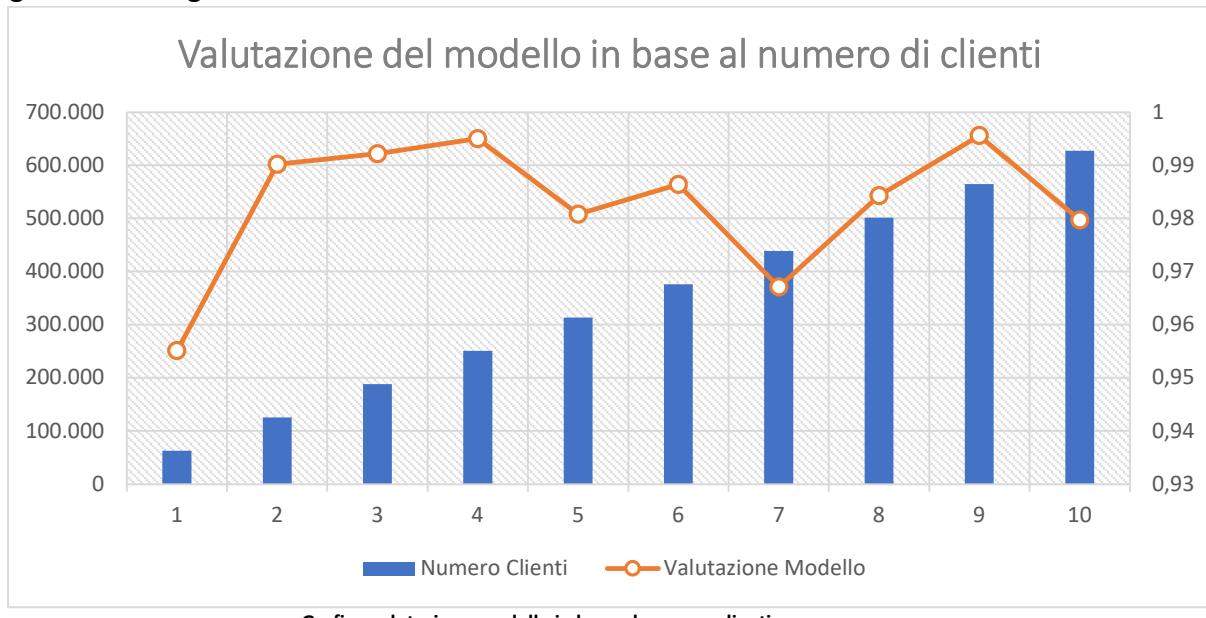
I valori indicati rappresentano la valutazione massima ottenuta dai sistemi di raccomandazione addestrati e valutati, per ognuno di essi aumentando il numero dei clienti sui quali viene addestrato il modello le prestazioni dello stesso possono diminuire questa è una caratteristica intrinseca dei dati acquisiti e quindi del dominio applicativo analizzato.

Nel report riportato di seguito viene mostrata la valutazione del modello definitivo eseguita su di un numero crescente in modo graduale di clienti.

Numero Clienti	Valutazione Modello
62.693	0,9551
125.386	0,9902
188.079	0,9922
250.772	0,9951
313.465	0,9808
376.158	0,9864
438.851	0,9671
501.544	0,9843
564.237	0,9956
626.929	0,9797

Prestazione del modello in base al numero di clienti

Dal report mostrato si nota l'aumento progressivo del numero dei clienti mentre la valutazione del modello nonostante cambi rimane comunque superiore allo 0,9 (90 %). Il grafico che segue mostra l'andamento delle due variabili.

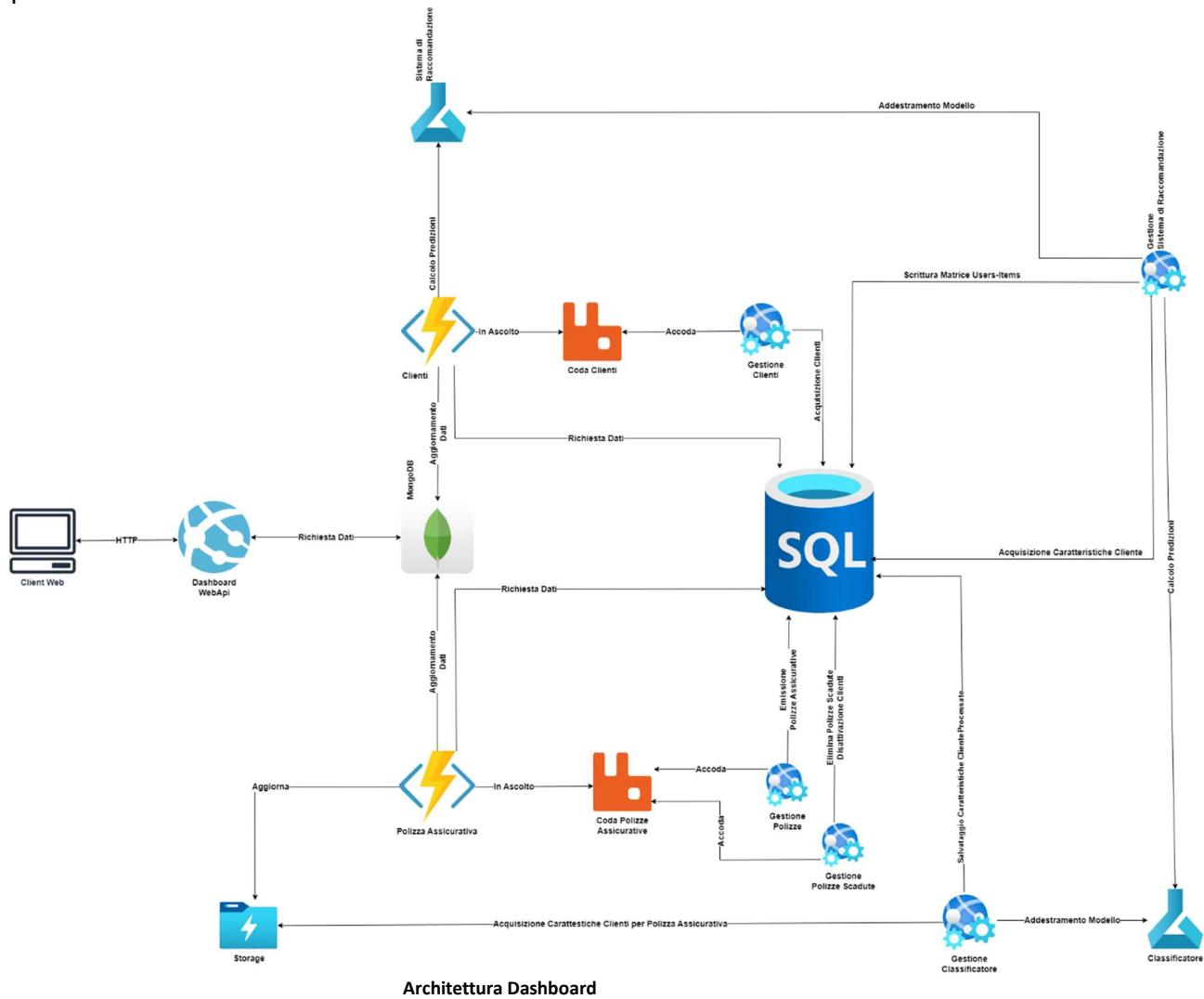


L'approccio ibrido utilizzando un classificatore multi-classe per determinare la probabilità che un cliente possa sottoscrivere in futuro una polizza combinato all'applicazione di un bias casuale mi ha permesso di ottenere un modello di raccomandazione con buone prestazioni.

Per valutare effettivamente l'efficacia di un sistema di raccomandazione è indispensabile utilizzarlo in un contesto reale, *anche su di un numero limitato di utenti o agenzie per verificare che i suggerimenti del sistema siano d'interesse per il cliente finale.*

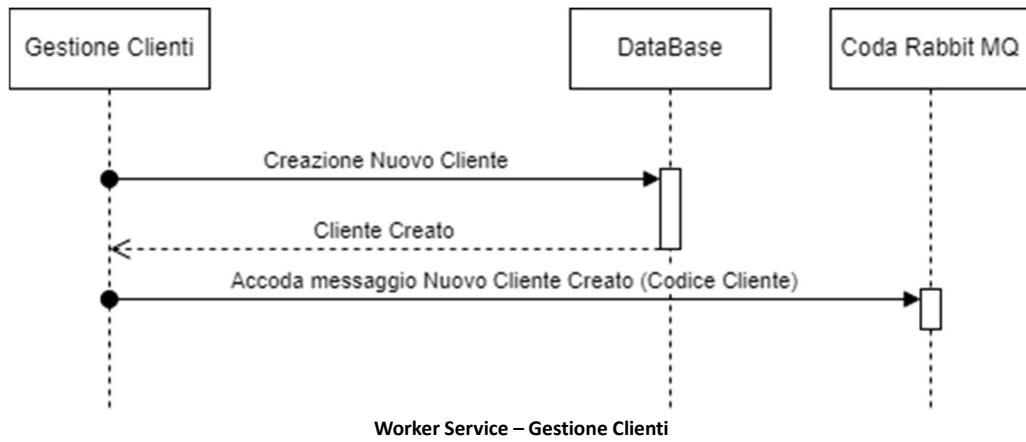
## 5. Architettura

L'immagine mostra l'architettura implementata per la gestione della piattaforma, in seguito verranno descritti i flussi implementati relativi ai vari componenti che costituiscono la piattaforma.

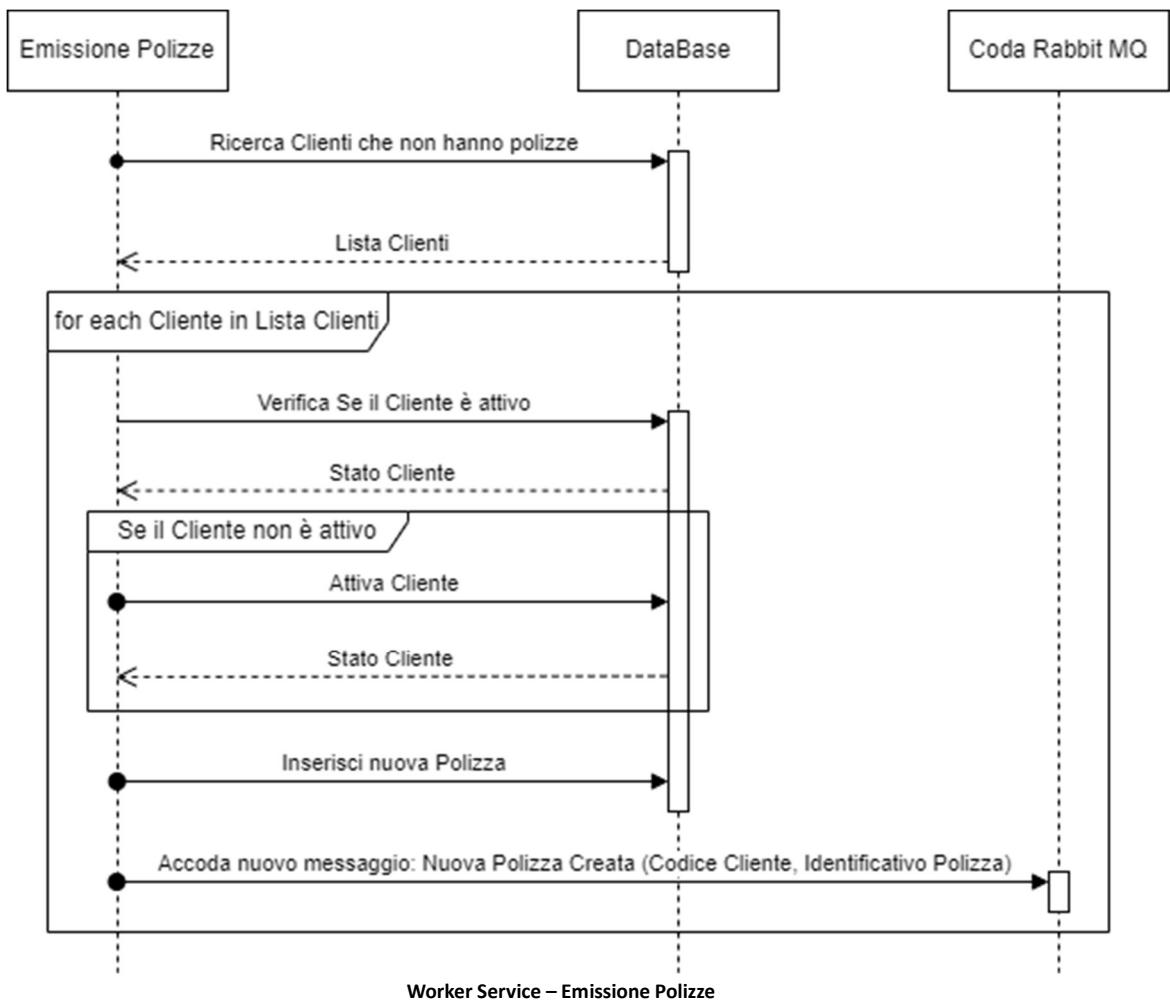


Avendo implementato una PoC (*Proof of Concept*) ci sono degli elementi che sono stati simulati attraverso dei Worker Service<sup>xxiv</sup>. I worker service implementati sono:

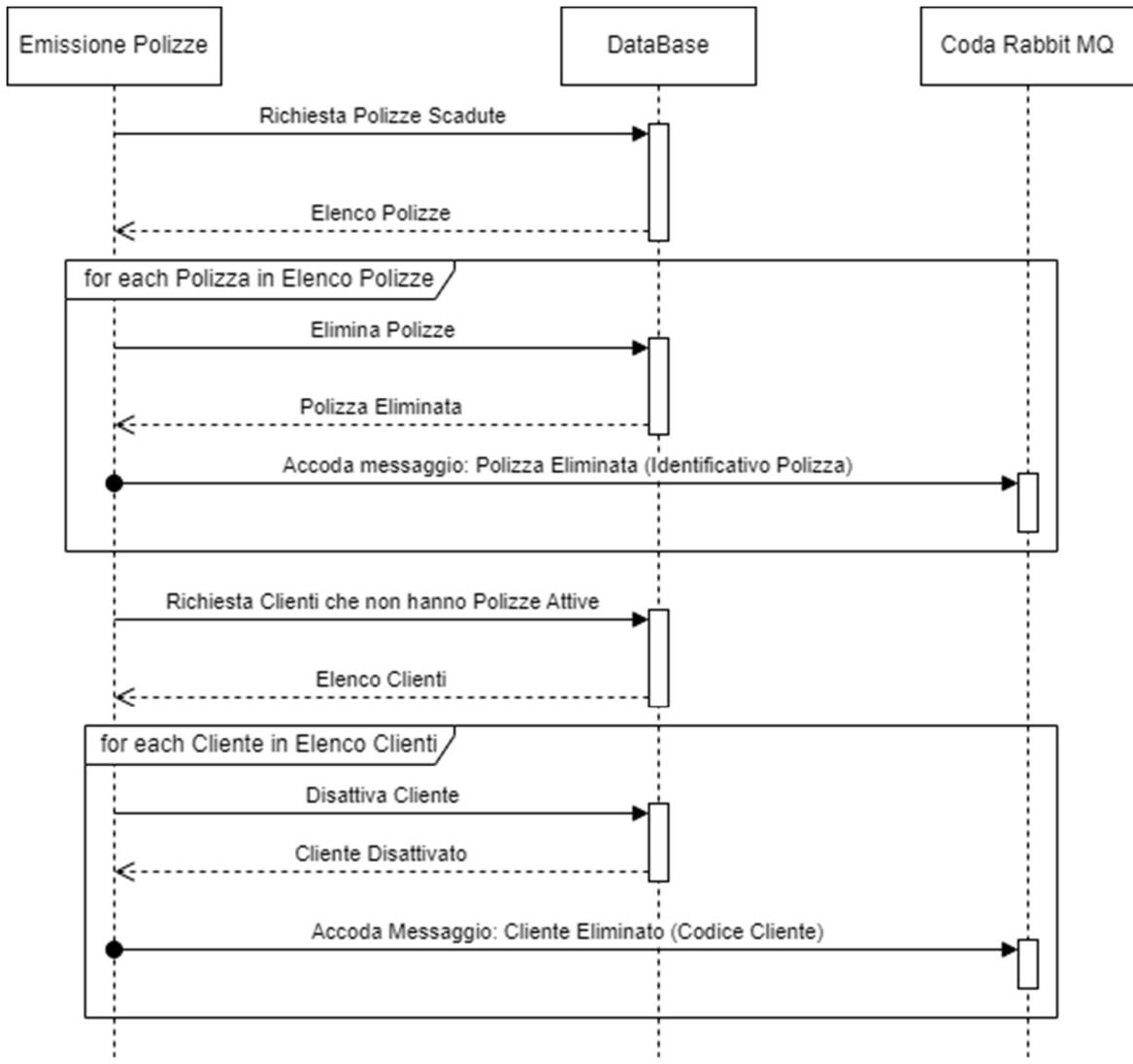
- **Gestione Clienti**: worker service che si occupa dell'acquisizione di nuovi clienti.



- **Gestione Polizze:** worker service che si occupa dell'emissione di nuove polizze sui clienti attivi e riattiva i clienti disattivati.

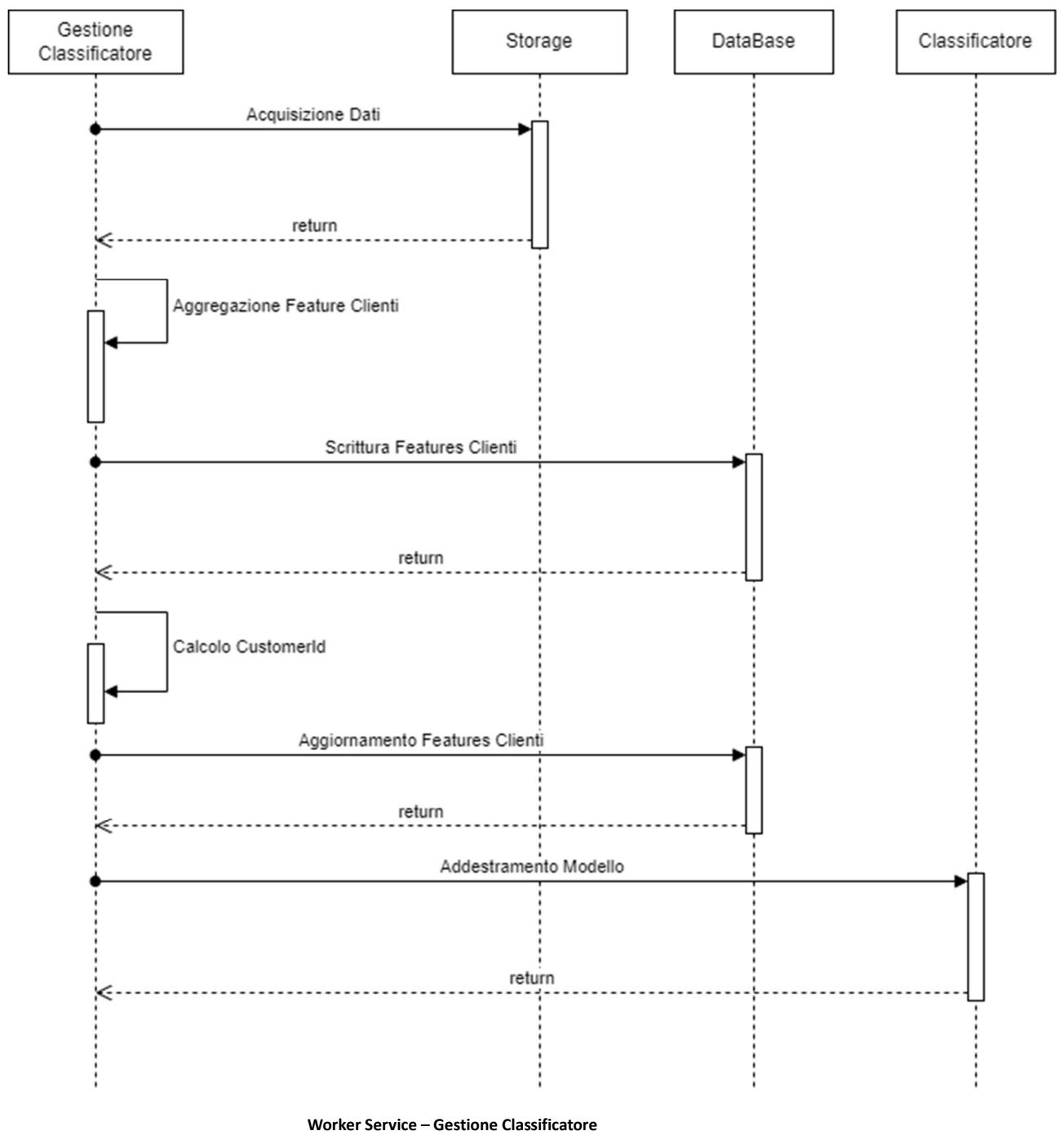


- **Gestione Polizze Scadute:** worker service che si occupa di cancellare le polizze scadute e la disattivazione dei clienti.



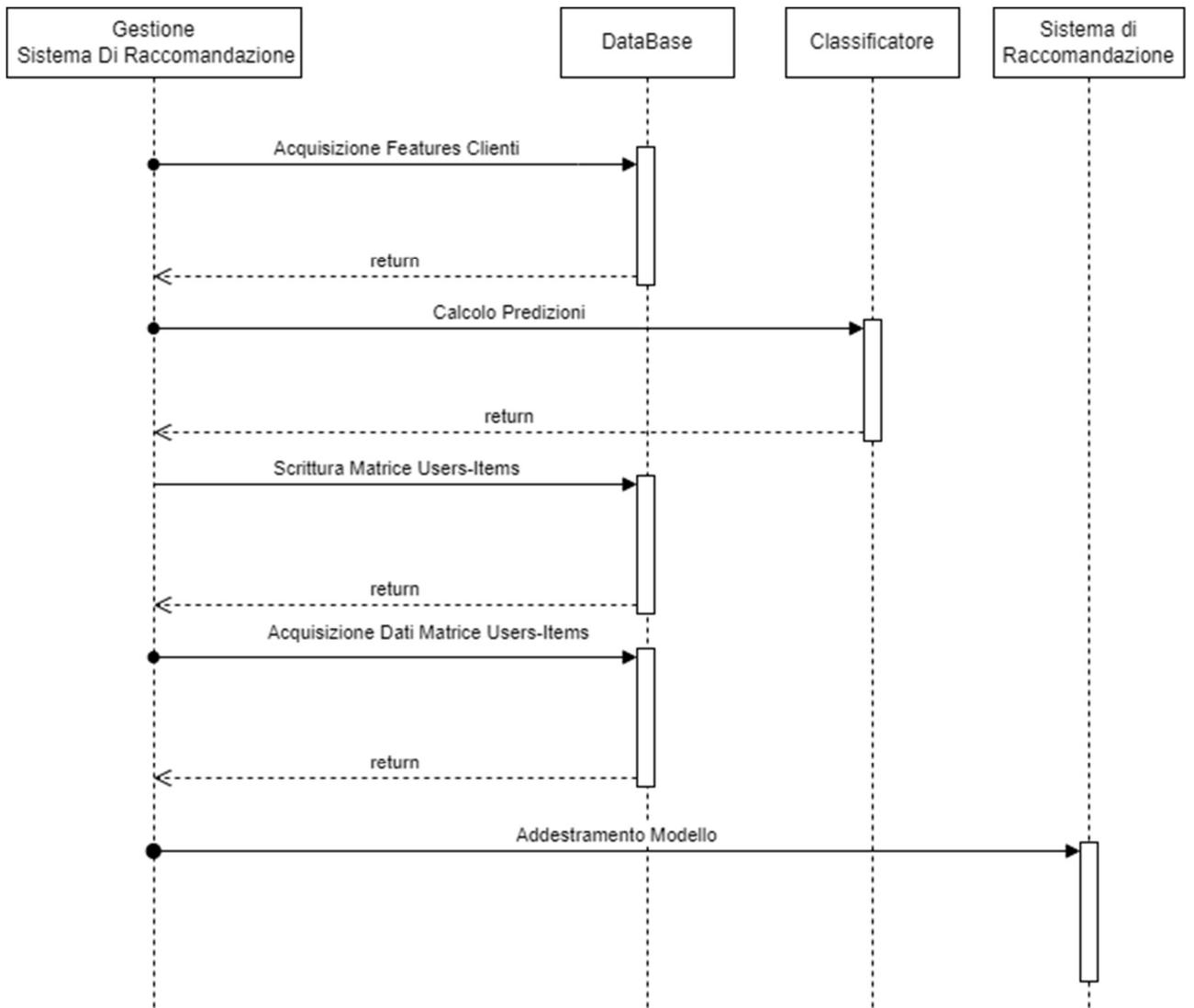
Worker Service – Eliminazione Polizze

- **Gestione Classificatore:** worker service che si occupa di acquisire le caratteristiche cliente e le polizze che ha sottoscritto dallo storage. Dopo procede a processarle ed a salvarle sul database (sql). Al termine del processo di acquisizione ed elaborazione dati provvede ad eseguire l'addestramento del modello di classificazione (*classificatore*).



Worker Service – Gestione Classificatore

- **Gestione Sistema di Raccomandazione:** worker service che si occupa di addestrare il sistema di raccomandazione, calcolando il rating che i vari clienti hanno dato alle polizze come score della previsione eseguita con il classificatore.

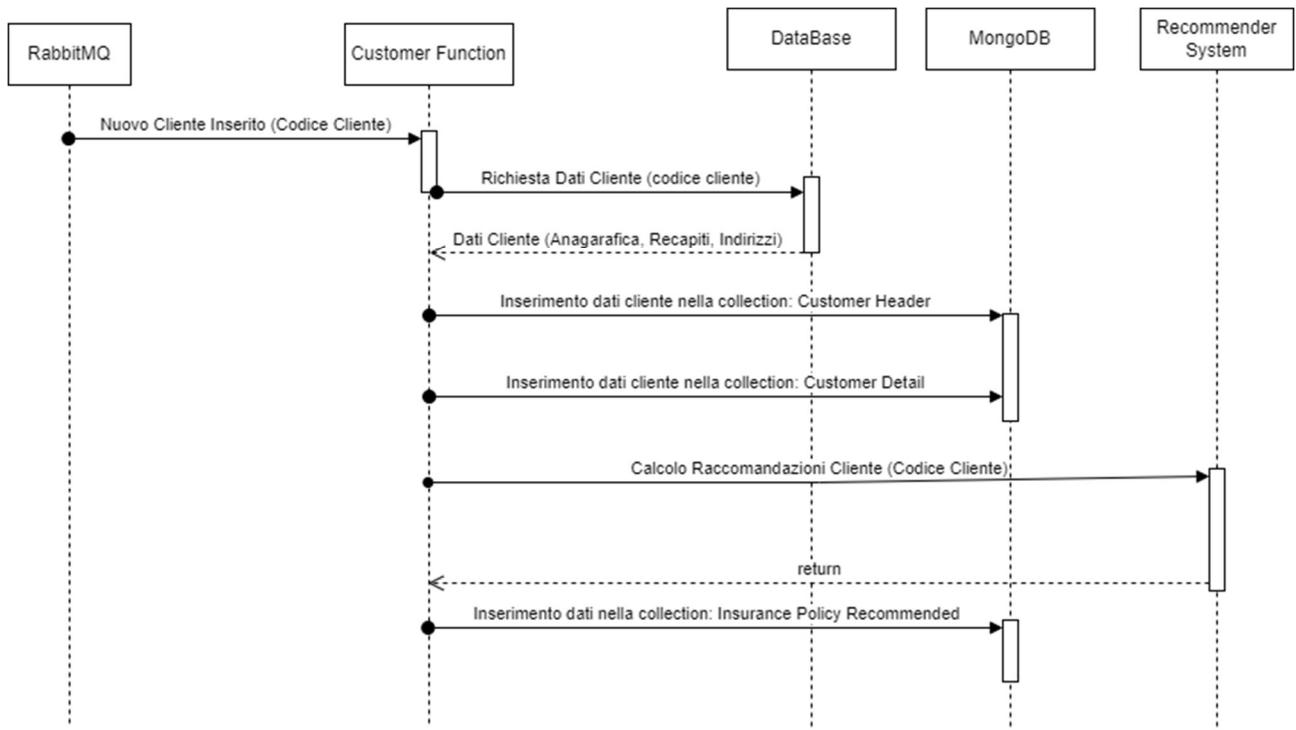


Worker Service – Gestione Sistema di Raccomandazione

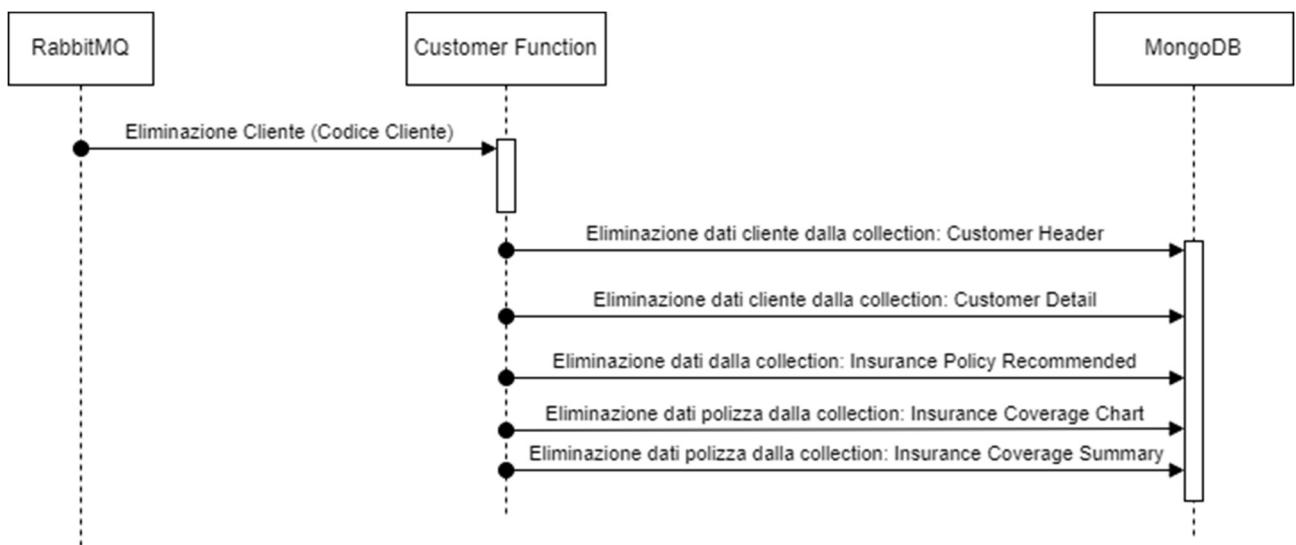
I worker service che simulano gli applicativi della piattaforma assicurativa dopo aver aggiornato i dati sul database provvedono ad inviare un messaggio su di una coda RabbitMQ<sup>xxv</sup>, i messaggi inviati vengono gestiti da processi serverless<sup>xxvi</sup> in ascolto sulla coda, i quali provvedono ad aggiornare le varie viste presenti su MongoDB i processi in questione sono stati implementati tramite delle Azure Function<sup>xxvii</sup>. I messaggi gestiti sono:

- Inserimento nuovo cliente
- Eliminazione cliente
- Inserimento nuova polizza
- Eliminazione polizza

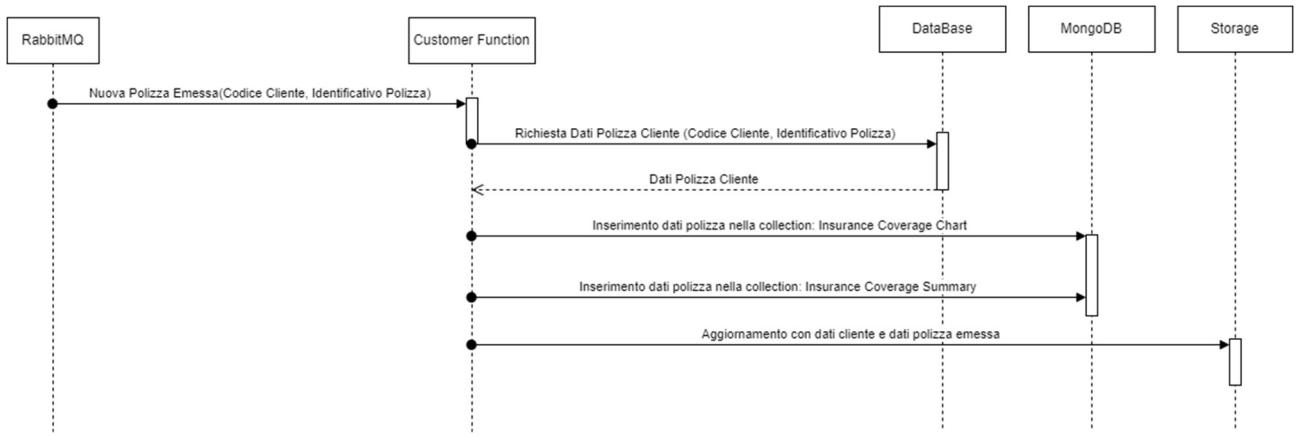
Di seguito verranno illustrati i diagrammi che descrivono il flusso gestito dalle Azure Function.



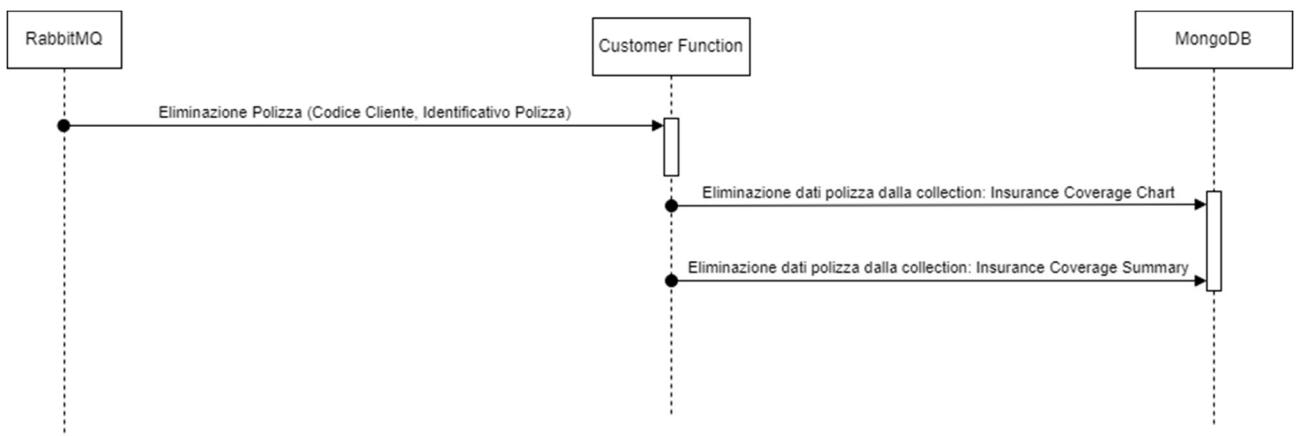
Azure Function – Inserimento Nuovo Cliente



Azure Function – Eliminazione Cliente



Azure Function – Inserimento Nuova Polizza

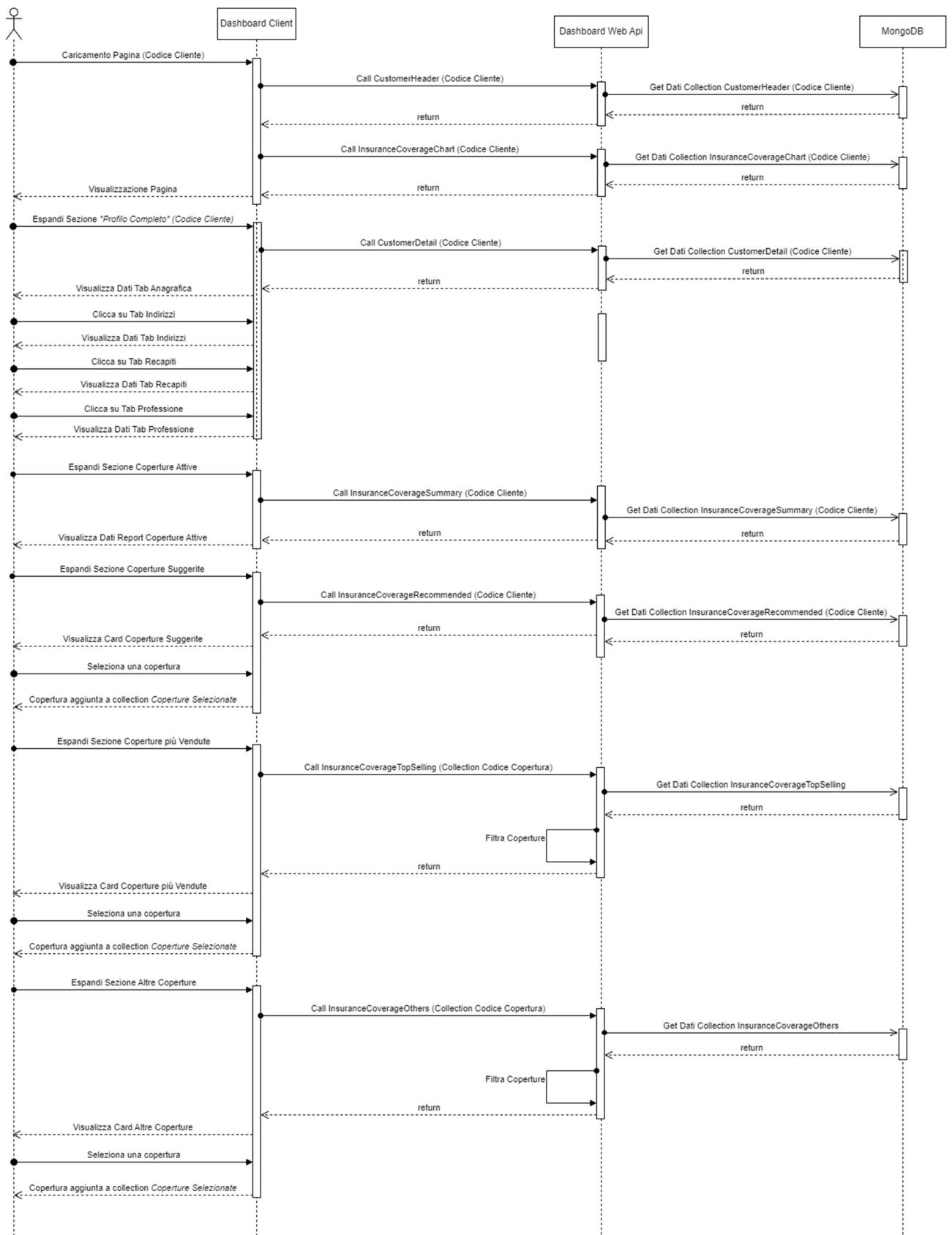


Azure Function – Eliminazione Polizza

L'agente assicurativo interagisce con la dashboard il cui front-end è stato implementato utilizzando *Angular*<sup>4</sup> mentre il back-end è stato implementato come una Web Api utilizzando *dotNET-7*<sup>5</sup> nel diagramma che segue viene mostrato il flusso d'interazione dell'utente con la dashboard.

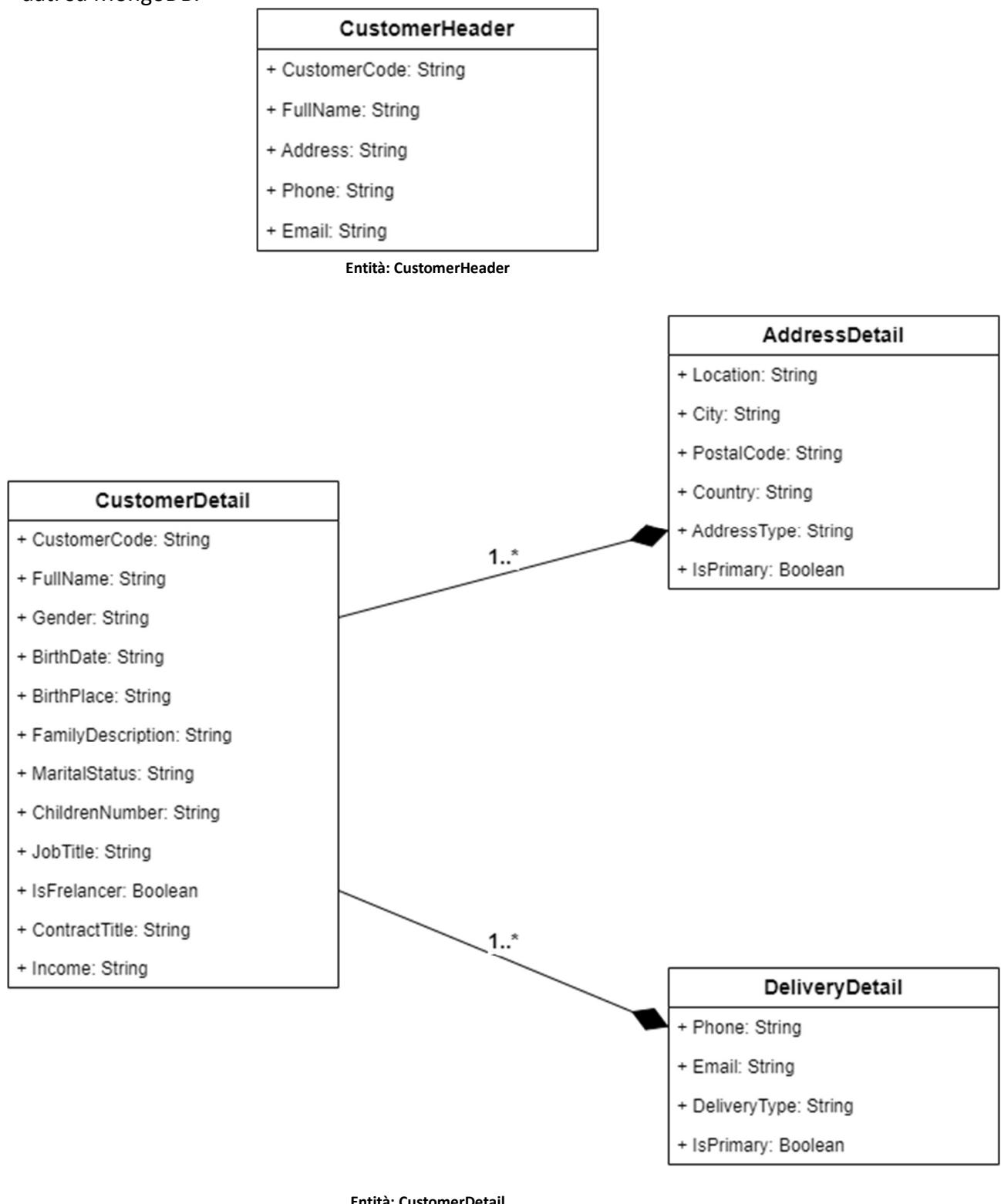
<sup>4</sup> [Angular](#)

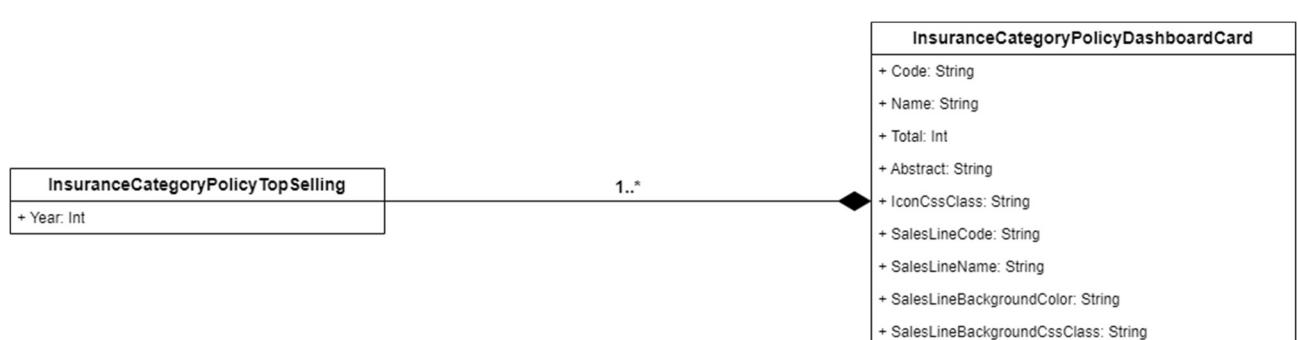
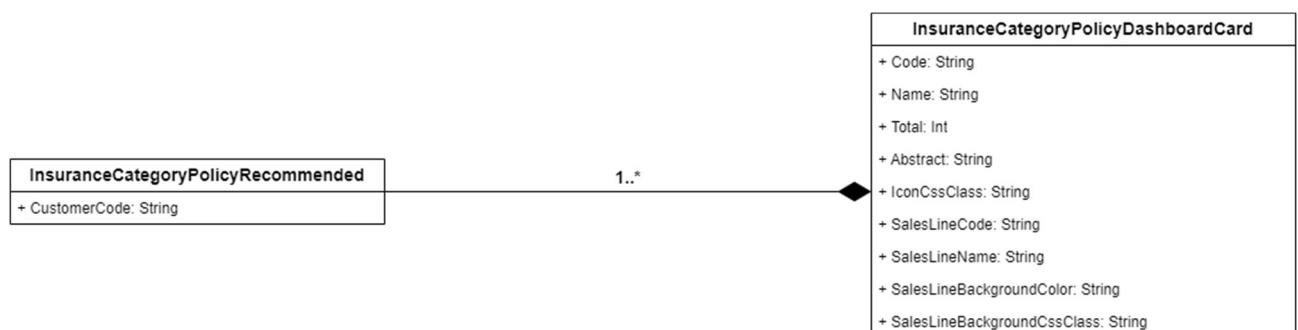
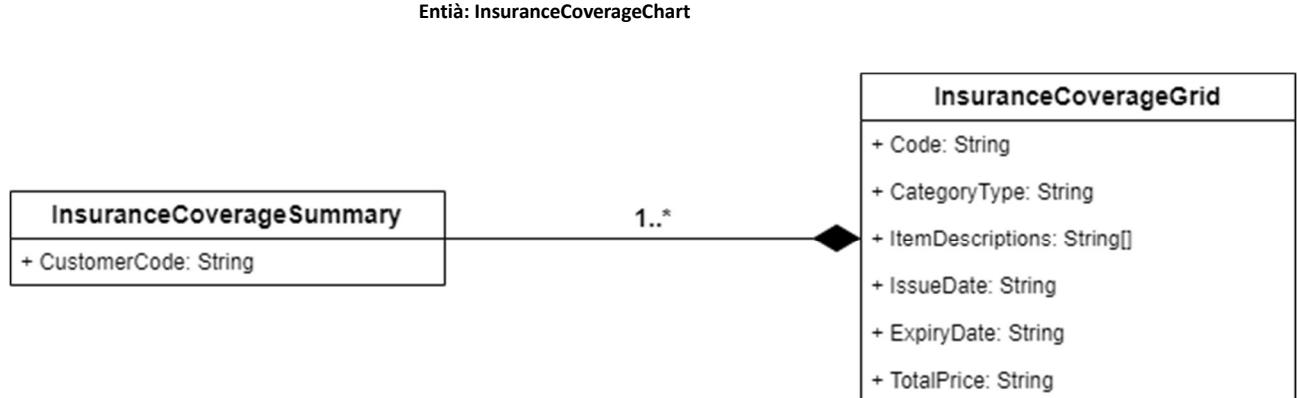
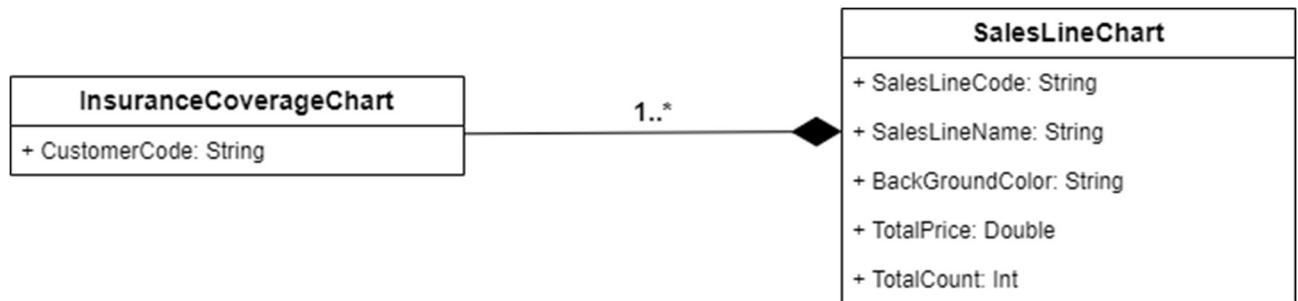
<sup>5</sup> [dotNET-7](#)



Interazione Dashboard

Nei diagrammi che seguono vengono mostrate le entità definite per la memorizzazione dei dati su MongoDB.





### **InsuranceCategoryPolicyDashboardCard**

- + Code: String
- + Name: String
- + Total: Int
- + Abstract: String
- + IconCssClass: String
- + SalesLineCode: String
- + SalesLineName: String
- + SalesLineBackgroundColor: String
- + SalesLineBackgroundCssClass: String

**Entità: InsuranceCategoryPolicyDashboardCard**

## 6. Conclusioni e sviluppi futuri

Il progetto presentato vuole fornire un'idea del supporto che l'impiego di tecnologie come il machine learning possono fornire al settore assicurativo.

In questo caso ho implementato un sistema di raccomandazione utilizzando un approccio collaborativo semi-supervisionato.

Il sistema implementato lavora off-line e i suggerimenti calcolati per il cliente vengono salvati su *MongoDB*: in questo modo l'utente della dashboard ha un'applicazione reattiva, dinamica e aggiornata realtime. Inoltre, evitando di calcolare le raccomandazioni ad ogni richiesta dell'utente, si evita un consumo di risorse il più delle volte superfluo. Per l'implementazione del sistema di raccomandazione è fondamentale addestrare il classificatore con una notevole base di dati eterogenea e consistente in modo da ottenere un sistema che accresca sempre più la sua accuratezza. L'approccio semi-supervisionato rispetto all'approccio supervisionato riduce le prestazioni del sistema in quanto vengono utilizzati dei valori che non sono stati acquisiti dagli utenti (*es. tramite intervista o questionario*) ma calcolati da un algoritmo di apprendimento automatico.

Gli sviluppi futuri che possono essere eseguiti possono essere molteplici.

- **Ampliare le features:** inserendo i dati dei beni assicurati, le garanzie acquistate, dati estratti da questionari compilati dal cliente (*es. questionario sulla salute*), solvibilità del cliente e metodi di pagamento utilizzati. In questo modo il classificatore, collezionando una mole di dati maggiore, fornirà delle predizioni più accurate. Se le features sono troppe da gestire è possibile applicare un algoritmo di clustering in modo da creare gruppi di clienti omogenei. Questa riduzione però potrebbe ridurre le prestazioni, quindi in fase di analisi e test dei modelli bisogna valutare se è il caso di applicare un algoritmo di clustering oppure utilizzare meno features in modo da ridurre la frammentazione dei clienti naturalmente.
- **Utilizzare un approccio ibrido:** sarebbe possibile creare una base di conoscenza assicurativa indicando le peculiarità di ogni polizza, *es. garanzie obbligatorie, garanzie facoltative, vincoli e clausole, modalità di pagamento ammesse, caratteristiche dei beni che possono essere assicurati*. In questo modo sarebbe possibile calcolare la similarità tra i prodotti assicurativi venduti, *utilizzando il TF-IDF descritto in precedenza*, integrando l'algoritmo collaborativo implementato.
- **Creazione di pacchetti assicurativi:** nella soluzione proposta l'algoritmo calcola la probabilità che un cliente possa sottoscrivere o meno una determinata polizza ed in seguito calcola i suggerimenti per il cliente. Inserendo le garanzie acquistate fra le caratteristiche del cliente ed implementando la base di conoscenza in modo da rappresentare le relazioni presenti fra le garanzie ed i vincoli esistenti sulle garanzie, possono essere creati dei pacchetti assicurativi su misura per ciascun cliente. Si potrebbe inoltre calcolare la probabilità che un cliente possa acquistare una determinata garanzia in futuro. Il sistema, quindi, andrebbe a suggerire al cliente dei pacchetti assicurativi personalizzati composti dalla polizza che si suggerisce di sottoscrivere indicando le garanzie che possono essere acquistate.

- 
- <sup>i</sup> [Recommender Systems](#), Prem Melville and Vikas Sindhwani
- <sup>ii</sup> [TF-IDF — Term Frequency-Inverse Document Frequency — LearnDataSci](#), Fatih Karabiber,
- <sup>iii</sup> [Stop word - Wikipedia](#), Wikipedia
- <sup>iv</sup> [Getting Started With Embeddings \(huggingface.co\)](#), Omar Espejel
- <sup>v</sup> [Coseno - Wikipedia](#), Wikipedia
- <sup>vi</sup> John S. Breese, David Heckerman, and Carl Kadie. Empirical analysis of predictive algorithms for collaborative filtering. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Madison, WI, July 1998
- <sup>vii</sup> Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.
- <sup>viii</sup> Daniel Billsus and Michael J. Pazzani. Learning collaborative information filters. In Proceedings of the Fifteenth International Conference on Machine Learning (ICML-98), pages 46–54, Madison, WI, 1998. Morgan Kaufmann
- <sup>ix</sup> [MAE - Mean Absolute Error](#)
- <sup>x</sup> [RDF - Resource Description Framework](#)
- <sup>xi</sup> [OWL - Web Ontology Language](#)
- <sup>xii</sup> [Ontologia Informatica](#)
- <sup>xiii</sup> [Neo4j - Docs](#)
- <sup>xiv</sup> [Overfitting](#)
- <sup>xv</sup> [Cold Start Problem \(recommender system\)](#)
- <sup>xvi</sup> [ML .NET](#)
- <sup>xvii</sup> [R-Squared \(R<sup>2</sup>\)](#)
- <sup>xviii</sup> [Mean absolute error](#)
- <sup>xix</sup> [Mean squared error](#)
- <sup>xx</sup> [Root mean square deviation](#)
- <sup>xxi</sup> [Matrix factorization \(recommender systems\)](#)
- <sup>xxii</sup> Prem Melville and Raymond J. Mooney and Ramadass Nagarajan. Content-Boosted Collaborative Filtering for Improved Recommendations
- <sup>xxiii</sup> [Bayesian Neural Networks](#)
- <sup>xxiv</sup> [Worker Service](#)
- <sup>xxv</sup> [RabbitMQ](#)
- <sup>xxvi</sup> [Serverless](#)
- <sup>xxvii</sup> [Azure Functions](#)