Sommario

[Introduzione al problema 1](#_Toc165882307)

[Analisi funzionale 4](#_Toc165882308)

[Profilo assicurativo 4](#_Toc165882309)

[Coperture assicurative 6](#_Toc165882310)

[Coperture selezionate 7](#_Toc165882311)

[I sistemi di raccomandazione 8](#_Toc165882312)

[Sistemi collaborativi 10](#_Toc165882313)

[Sistemi memory Based (neighborhood based) 11](#_Toc165882314)

[Sistemi model based 13](#_Toc165882315)

[Sistemi ibridi 14](#_Toc165882316)

[Valutazione di un sistema 14](#_Toc165882317)

[Architettura 15](#_Toc165882318)

[Considerazioni in ambito assicurativo 21](#_Toc165882319)

[Soluzione implementata 21](#_Toc165882320)

[Conclusioni e sviluppi futuri 27](#_Toc165882321)

# 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.

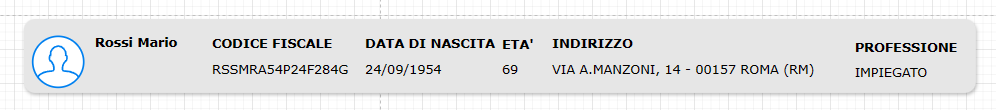
L’attuale sistema emissivo delle polizze assicurative risulta essere farraginoso, di poco impatto e anonimo. L’immagine mostra la pagina utilizzata dagli agenti per l’emissione delle polizze.

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

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.



Dati Cliente

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.

Immagine che contiene testo, schermata, Policromia, software

Descrizione generata automaticamente

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 emersi 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.

# Analisi funzionale

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

La sua progettazione prevede ruota intorno al cliente. 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:

Immagine che contiene testo, schermata, software, numero

Descrizione generata automaticamente

Cruscotto cliente

## 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 delle coperture assicurative, il nominativo, ed i suoi recapiti principali. L’immagine mostra un esempio appena descritto.

Immagine che contiene testo, schermata, software, Carattere

Descrizione generata automaticamente

Profilo assicurativo del cliente

Nella schermata visualizzata sono mostrati:

* Il nominativo del cliente nell’intestazione della pagina.
* Sulla sinistra un grafico riassuntivo delle coperture attive, *in questo caso due coperture per i veicoli e quattro per la famiglia.*
* 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 visualizzata la il profilo completo del cliente. È prevista una scheda per ogni tipologia di dato *anagrafica, indirizzo, recapiti,* professione.

Immagine che contiene testo, numero, Carattere, schermata

Descrizione generata automaticamente

Profilo completo cliente

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:

Immagine che contiene testo, schermata, software, Pagina Web

Descrizione generata automaticamente

Coperture assicurative

Il report mostra tre polizze:

* Polizza Auto
* Polizza Animale Domestico
* Polizza Casa

Per ogni copertura viene mostrato:

* Identificativo polizza
* Tipologia polizza
* Dettaglio bene assicurato
* Data stipulazione contratto
* Data scadenza contratto
* Importo premio totale

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

## 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 suggerite direttamente all’utente calcolate dal sistema di raccomandazione.
* **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*, la quale contiene i suoi dati:

* Icona del prodotto
* Nome del prodotto
* Categoria del prodotto
* Descrizione del prodotto
* Spunta attraverso la quale è possibile selezionare o deselezionare il prodotto

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

Immagine che contiene testo, Carattere, linea, schermata

Descrizione generata automaticamente  
Card relative alle coperture assicurative

## Coperture selezionate

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

Essa contiene dei segnaposti 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.

Immagine che contiene testo, schermata, Carattere, Marchio

Descrizione generata automaticamente

Coperture selezionate

# I sistemi di raccomandazione

L’obbiettivo di un sistema di raccomandazione[[1]](#endnote-1) 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*.

Immagine che contiene numero, schermata, parole crociate, calendario

Descrizione generata automaticamente

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 **rij** *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’obbiettivo 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.

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Tipologia sistemi di raccomandazione

## 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

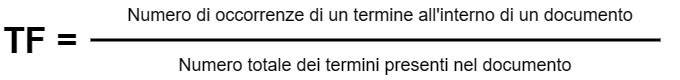
I campi proposti rappresentano un sotto insieme di una possibile base di conoscenza reale; ogni campo a sua volta può essere collegato ad altre basi di conoscenza. Il genere potrebbe essere descritto come:

* 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.

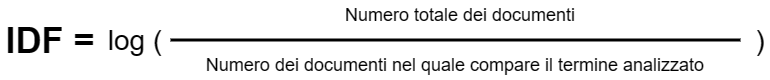
Dopo aver creato e popolato la base di conoscenza bisogna misurare la similarità tra gli elementi; essa viene calcolata utilizzando il **TF-IDF[[2]](#endnote-2)**, Il TF-IDF è costituito dal *TF - Term Frequency* e dall’*IDF - Inverse Document Frequency*.

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”*.



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”*.



Formula per il calcolo di IDF – Inverse Document Frequency

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

Immagine che contiene testo, Carattere, bianco, Elementi grafici

Descrizione generata automaticamente

Formula per il calcolo di TF-IDF

L’utilizzo del TF-IDF permette di non dover eliminare le *“stop words*[[3]](#endnote-3)*”* 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**[[4]](#endnote-4), la similarità viene misurata calcolando il coseno[[5]](#endnote-5) 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)*.

## 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.

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente

Suddivisione – sistemi collaborativi

### 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 *wa,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*.

Immagine che contiene testo, Carattere, linea, bianco

Descrizione generata automaticamente

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.
* ru,i: è la valutazione dell’utente *u* relativo all’item *i*.
* ra,i: è la valutazione dell’utente attivo *a* rispetto all’item *i*.
* : indica la media delle valutazioni eseguite dall’utente *u*.
* ****: 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.

Immagine che contiene testo, Carattere, linea, bianco

Descrizione generata automaticamente

Formula per il calcolo dele predizioni

La formula relativa alle predizioni è definita come segue:

* pa,i: è la previsione per l’utente *a* relativa all’elemento *i*.
* wa,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. Ci sono diverse alternative per il calcolo della similarità, anche se studi empirici[[6]](#endnote-6) hanno dimostrato che il coefficiente di correlazione di Pearson offre prestazioni migliori.

L’approccio basato sul vicinato risulta essere poco scalabile in quanto la ricerca di utenti simili ha un’elevata complessità computazionale. Linden[[7]](#endnote-7) 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:

Immagine che contiene testo, Carattere, linea, bianco

Descrizione generata automaticamente

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*.
* ru,i: è la valutazione dell’utente *u* relativo all’item *i*.
* : è la media delle valutazioni degli utenti rispetto all’item *i*.
* : 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:

Immagine che contiene Carattere, testo, bianco, linea

Descrizione generata automaticamente

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

### Sistemi model based

I sistemi basati sul modello forniscono le raccomandazioni in base alle statistiche sulle preferenze degli utenti. Un prima approccio[[8]](#endnote-8) 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[[9]](#footnote-1) e le valutazioni dell’utente come le labels[[10]](#footnote-2).

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)* *wu, hi ∈ ℜk* , lungo *k* dimensioni latenti *(nascoste)*. Queste caratteristiche *(features)* approssimano la valutazione dell’utente *u* relativa all’elemento *i* attraverso il prodotto *wuThi*. *ru,i* rispettando la funzione di costo *(loss function)[[11]](#endnote-9)*.

La scelta predefinita per la funzione di costo *(loss function)* è la *squared loss[[12]](#endnote-10)*, l’obbiettivo è minimizzare la funzione

Immagine che contiene testo, Carattere, bianco, calligrafia

Descrizione generata automaticamente

Formula Squared Loss Function

Gli elementi presenti nella *squared loss function* sono:

* W = [w1…wn]T : è una matrice di rango *n \* k*.
* H = [h1…hm] : è una matrice di rango m \* k.
* L : identifica l’insieme di coppie *user-item* per cui il *rating[[13]](#footnote-3)* è noto.

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 si può ridurre il problema fattorizzando la matrice *R* utilizzando la decomposizione a valori singolari troncata *(truncated SVD)[[14]](#endnote-11)* di *R*. Utilizzando questo metodo *R* equivale al prodotto delle matrici U, D, VT *(la trasposta di V)*, *R = UDVT*. Successivamente si impostano le matrici *W* e *H*, rispettivamente. *W* è il risultato del prodotto fra le prime *k* colonne di U *(Uk)* e la radice quadrata della matrice D troncata *(Dk1/2)*, mentre *H* è il risultato del prodotto tra la radice quadrata della matrice D troncata *(Dk1/2)* e le prime *k* righe della trasposta della matrice V *(*VkT*)*, dove *Uk, Dk, VkT* contengono i *k* triplete singolari maggiori di R.

## 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.

## Valutazione di un sistema

Per valutare un sistema di apprendimento automatico, vengono utilizzate delle metriche di valutazione, le metriche principalmente utilizzate sui sistemi di raccomandazione sono il *supporto* e la *confidenza*.

# Architettura

L’architettura implementata per l’applicazione Web da fornire come supporto agli agenti assicurativi viene visualizzata nell’immagine

Immagine che contiene testo, diagramma, schermata, Piano

Descrizione generata automaticamente

Architettura Dashboard

Avendo implementato una PoC *(Proof of Concept)* ci sono degli elementi che sono stati simulati attraverso dei Worker Service[[15]](#endnote-12). I worker service implementati sono:

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

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente  
Worker Service – Gestione Clienti

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

Immagine che contiene testo, schermata, Parallelo, diagramma

Descrizione generata automaticamente

Worker Service – Emissione Polizze

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

Immagine che contiene testo, Parallelo, schermata, numero

Descrizione generata automaticamente  
Worker Service – Eliminazione Polizze

* **Gestione Classificatore**: worker service che si occupa di acquisire le caratteristiche cliente e le corrispondenti polizze emesse dallo storage e dopo averle processate le salva sul database (sql) ed infine addestra il modello di classificazione.

Immagine che contiene testo, diagramma, Parallelo, linea

Descrizione generata automaticamente  
Worker Service – Gestione Classificatore

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

Immagine che contiene testo, Parallelo, diagramma, linea

Descrizione generata automaticamente  
Worker Service – Gestione Sistema di Raccomandazione

I worker service descritti nei diagrammi dopo aver aggiornato i dati sul database provvedono ad inviare un messaggio su di una coda RabbitMQ[[16]](#endnote-13), i messaggi inviati vengono gestiti da un processo serverless[[17]](#endnote-14) in ascolto sulla coda, il quale provvede ad aggiornare le varie viste presenti su MongoDB il processo in questione è stato implementato tramite un’Azure Function[[18]](#endnote-15). 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.

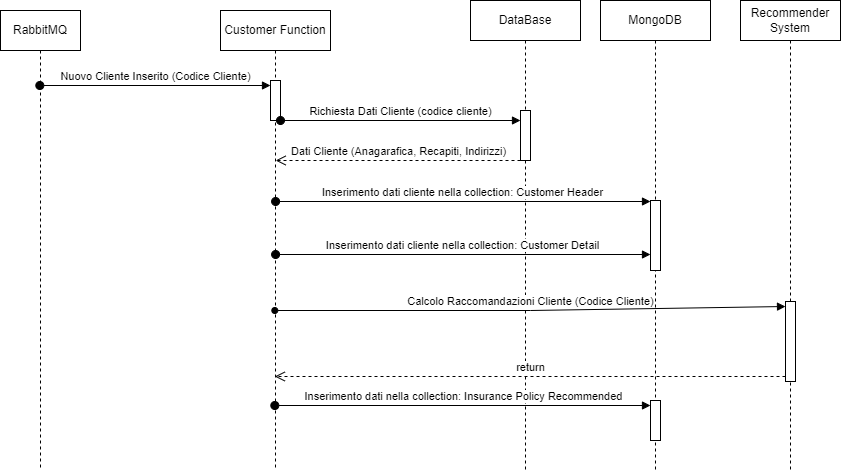
  
Azure Function – Inserimento Nuovo Cliente

Immagine che contiene testo, schermata, linea, Carattere

Descrizione generata automaticamente  
Azure Function – Eliminazione Cliente

Immagine che contiene testo, linea, diagramma, schermata

Descrizione generata automaticamente  
Azure Function – Inserimento Nuova Polizza

Immagine che contiene testo, schermata, linea, diagramma

Descrizione generata automaticamente  
Azure Function – Eliminazione Polizza

## Considerazioni in ambito assicurativo

Prima di descrivere le implementazioni previste per 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.
* **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.
* **Vincoli**: quando si sottoscrive un’assicurazione ci sono diversi vincoli che vanno rispettati.
* **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.

Nella soluzione implementata sono stati affrontati l’affidabilità e l’assenza di rating esplicito. L’affidabilità è stata affrontata testando diversi modelli su diverse dimensioni di dati fino a quando si è ottenuto un modello con un elevato score.

L’assenza di rating esplicito è stato affrontato implementando un sistema semi-supervisionato dove le valutazioni sono state calcolate in precedenza da un modello di classificazione il quale provvede a determinare qual è la probabilità che un cliente sottoscriva una determinata polizza lo score ottenuto rappresenta la valutazione impostata per il sistema di raccomandazione.

## Soluzione implementata

Il sistema di raccomandazione implementato segue due passi, nel primo passo viene addestrato un modello di classificazione per determinare in base alle caratteristiche del cliente *(Features)* quale polizza potrebbe sottoscrivere. Nel secondo passo viene popolata la matrice *users-items* con l’identificativo del cliente *(UserId)*, l’identificativo della polizza *(ItemId)*, come rating viene utilizzato lo score ottenuto dalla predizione con il classificatore. Dopo aver compilato la matrice *users-items* viene addestrato il sistema di raccomandazione.

I dati utilizzati per definire le *features* del cliente sono dati *anonimi* dei clienti assicurativi con l’indicazione della polizza sottoscritta, nella tabella sono mostrati i dati che sono stati estratti e la loro tipologia:

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

Dati Cliente presenti nello Storage

Per proseguire con l’implementazione è necessario analizzare i dati acquisiti per capire come siano distribuiti, nella tabella e grafico viene mostrata la distribuzione dei dati:

|  |  |
| --- | --- |
| **Nome Polizza** | **Numero Clienti** |
| R.C.A. | 574047 |
| INFORTUNI | 64399 |
| MULTIGARANZIA ABITAZIONE | 2415 |
| MALATTIA | 768 |
| R.C. DIVERSI | 78 |
| A.R.D. | 22 |
| GLOBALE FABBRICATI | 8 |
| INCENDIO/FURTO | 4 |
| TUTELA GIUDIZIARIA | 2 |

Tabella con Distribuzione dei dati

Grafico che mostra la distribuzione delle polizze acquistate dai clienti

Come si può notare la distribuzione dei dati è fortemente sbilanciata, in questo caso non sarebbe possibile creare un sistema di raccomandazione affidabile in quanto se andiamo a rappresentare la matrice *users-items* indicando le caratteristiche dei clienti e come valutazione:

* 1 – se il cliente ha sottoscritto la polizza.
* 0 – se il cliente non ha sottoscritto la polizza.

Avremo come risultato una matrice sparsa fortemente sbilanciata verso le polizze *R.C.A.* e quindi il sistema di raccomandazione ritornerebbe come suggerimento sempre *R.C.A.* con una probabilità del 100%.

Quindi sono state definite delle funzioni che vanno a trasformare i dati acquisiti in modo da creare una maggiore similarità tra gli utenti. Nella tabella che segue vengono indicati quali sono i campi che sono stati elaborati ed il loro risultato

|  |  |  |
| --- | --- | --- |
| **Campo di Partenza** | **Campi di Arrivo** | **Descrizione Processo** |
| Data di Nascita | Mese di nascita | Estrazione del mese di nascita dalla data |
| Data di Nascita | Anno di nascita | Estrazione dell’anno di nascita dalla data |
| Stato Civile | Stato Civile | Non viene eseguita alcuna elaborazione |
| Stato Civile | Single | Viene valorizzato con true per i valori: *(Celibe/Nubile, Divorziato, Vedovo)* con false per i valori: *(Coniugato, Convivente)* |
| Coniuge a Carico | Coniuge a Carico | Conversione in booleano |
| Numero Figli | Numero Figli | Non viene eseguita alcuna elaborazione |
| Numero Figli a Carico | Numero Figli a Carico | Non viene eseguita alcuna elaborazione |
| Professione | Professione | Non viene eseguita alcuna elaborazione |
| Reddito (RAL) | Fascia di Reddito | I valori di conversione verranno indicati nella tabella successiva |
| Tipo Impiego | Tipo Impiego | Non viene eseguita alcuna elaborazione |
| Tipo Impiego | Freelancer | Viene valorizzato con true se Tipo Impiego vale: *Libero professionista,* viene valorizzato con false se tipo impiego vale *Lavoratore dipedente* |
| Provincia di Residenza | Provincia di Residenza | Non viene eseguita alcuna elaborazione |
| Provincia di Residenza | Regione di Residenza | Viene estratta dal database la regione associata alla provincia di residenza |
| Nome Polizza | Nome Polizza | Non viene eseguita alcuna elaborazione |
| Nome Polizza | Identificativo Polizza | Estrazione dal database dell’identificativo della polizza stipulata |
| Nome Polizza | Codice Polizza | Estrazione dal database del codice della polizza stipulata |

Tabella di Conversione per le elaborazioni applicate

|  |  |  |
| --- | --- | --- |
| **Fascia Reddito** | **Limite Reddito Minimo** | **Limite Reddito Massimo** |
| Reddito Basso | 0 | 15000 |
| Reddito Medio Basso | 15001 | 28000 |
| Reddito Medio | 28001 | 55000 |
| Reddito Medio Alto | 55001 | 75000 |
| Reddito Alto | 75001 | 120000 |
| Reddito Molto Alto | 120001 | Nessun Limite |

Tabella di Conversione da Reddito (RAL) a Fascia Reddito

Dopo aver eseguito le varie elaborazioni, otteniamo i dati che verranno utilizzati per l’addestramento del classificatore. Nella tabella che segue i dati ottenuti:

|  |  |  |
| --- | --- | --- |
| **Campi di Arrivo** | **Tipologia Campo** | **Descrizione Campo** |
| Mese di nascita | Numerico | Mese di nascita del cliente |
| Anno di nascita | Numerico | Anno di nascita del cliente |
| Stato Civile | Stringa | Stato civile del cliente |
| Single | Booleano | Indica se lo stato civile del cliente indica che è single oppure in coppia |
| Coniuge a Carico | Booleano | Indica se il coniuge è a carico del cliente |
| Numero Figli | Numerico | Numero di figli del cliente |
| Numero Figli a Carico | Numerico | Numero di figli a carico del cliente |
| Professione | Stringa | Professione del cliente |
| Fascia di Reddito | Stringa | Indica la fascia di reddito a cui appartiene il cliente |
| Tipo Impiego | Stringa | Indica il tipo di lavoro svolto dal cliente |
| Freelancer | Booleano | Indica se il cliente lavora come libero professionista o lavoratore dipendente |
| Provincia di Residenza | Stringa | Indica la provincia di residenza del cliente |
| Regione di Residenza | Stringa | Indica la regione di residenza del cliente |
| Nome Polizza | Stringa | Indica il nome della polizza sottoscritta dal cliente |
| Identificativo Polizza | Numerico | Indica l’identificativo della polizza sottoscritta dal cliente |
| Codice Polizza | Stringa | Indica il codice della polizza sottoscritta dal cliente |

Campi utilizzati per l’addestramento del classificatore

Dopo aver eseguito la fase di aggregazione ed elaborazione dei dati il sistema provvede a calcolare il *CustomerId* questo è un valore che viene aggiunto alle *Features* ma viene ignorato in fase di addestramento del classificatore. Il *CustomerId* viene definito per identificare in modo univoco un cliente con date caratteristiche e successivamente utilizzarlo per popolare la matrice *users-items*. Il *CustomerId* viene calcolato partendo dal presupposto che un cliente con le stesse caratteristiche può aver stipulato polizze differenti, quindi vengono estratte tutte le *features* relative ai Clienti *(viene eseguita una distinct su tutte le features)* viene estratto l’ultimo *CustomerId* calcolato e per ogni cliente individuato viene incrementato di 1, al termine del processo aggiorna i record sul db. Nello pseudo-codice che segue viene mostrato l’algoritmo appena descritto:

Immagine che contiene testo, documento, schermata, Carattere

Descrizione generata automaticamente

Pseudo-codice per il calcolo del CustomerId

Dopo aver calcolato il *CustomerId* per tutti i clienti viene eseguito l’addestramento del classificatore e scelto l’algoritmo di classificazione che presenta le prestazioni migliori, l’algoritmo selezionato è stato il LbfgsLogisticRegressionBinaryTrainer*[[19]](#endnote-16)*. Nel report che segue vengono mostrate le prestazioni dei vari algoritmi che sono stati testati.

Dopo aver eseguito l’addestramento del classificatore si passa al popolamento della matrice users-items. Essa viene popolata calcolando la probabilità che un cliente sottoscriva una polizza per l’occorrenza *cliente-polizza* viene salvata la probabilità ottenuta dal classificatore. Nello pseudo-codice che segue viene mostrata l’algoritmo di popolazione della matrice users-items.

Immagine che contiene testo, schermata, Carattere, documento

Descrizione generata automaticamente  
Algoritmo di popolamento Matrice Users-Items

Come si può notare dall’esempio mostrato nell’immagine dopo aver eseguito l’algoritmo descritto la matrice *users-items* che ottengo è una matrice densa in quanto tutte le occorrenze per ogni user e per ogni item sono valorizzate.

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Esempio della matrice users-items

# 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 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 sono 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.

1. **Recommender Systems**, Prem Melville and Vikas Sindhwani [↑](#endnote-ref-1)
2. [**TF-IDF — Term Frequency-Inverse Document Frequency – LearnDataSci**](https://www.learndatasci.com/glossary/tf-idf-term-frequency-inverse-document-frequency/), Fatih Karabiber, [↑](#endnote-ref-2)
3. [**Stop word - Wikipedia**](https://en.wikipedia.org/wiki/Stop_word), Wikipedia [↑](#endnote-ref-3)
4. [**Getting Started With Embeddings (huggingface.co)**](https://huggingface.co/blog/getting-started-with-embeddings), Omar Espejel [↑](#endnote-ref-4)
5. [**Coseno - Wikipedia**](https://it.wikipedia.org/wiki/Coseno), Wikipedia [↑](#endnote-ref-5)
6. 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 [↑](#endnote-ref-6)
7. Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. IEEE Internet Computing,

   7(1):76–80, 2003. [↑](#endnote-ref-7)
8. 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 [↑](#endnote-ref-8)
9. Feature: le caratteristiche che identificano un elemento *(item)* [↑](#footnote-ref-1)
10. Label: l’etichetta valore su cui eseguire la predizione [↑](#footnote-ref-2)
11. [Loss Function](https://en.wikipedia.org/wiki/Loss_function) [↑](#endnote-ref-9)
12. Squared Loss, Matthew Bernstein 2017 [↑](#endnote-ref-10)
13. Rating: valutazione di un utente su di un elemento [↑](#footnote-ref-3)
14. [Decomposizione a valori singolari (SVD)](https://it.wikipedia.org/wiki/Decomposizione_ai_valori_singolari) [↑](#endnote-ref-11)
15. [Worker Service](https://www.w3.org/TR/service-workers/) [↑](#endnote-ref-12)
16. [RabbitMQ](https://en.wikipedia.org/wiki/RabbitMQ) [↑](#endnote-ref-13)
17. [Serverless](https://en.wikipedia.org/wiki/Serverless_computing) [↑](#endnote-ref-14)
18. [Azure Functions](https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview?pivots=programming-language-csharp) [↑](#endnote-ref-15)
19. [SdcaMaximumEntropyMulticlassTrainer](https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.sdcamaximumentropymulticlasstrainer?view=ml-dotnet) [↑](#endnote-ref-16)