Università Politecnica delle Marche

Facoltà di Ingegneria

Corso di Laurea Magistrale in Ingegneria Informatica e dell'Automazione

Tesi di Laurea

Progettazione e realizzazione di una campagna di data science per la segmentazione dei clienti e il calcolo di indicatori di rischio per un gruppo bancario

Design and implementation of a data science campaign to perform customer segmentation and compute risk indicators for a banking group

Relatore	Candidato
Prof. Domenico Ursino	Carlo Peroni

Anno Accademico 2019-2020

Indice

Elenco delle figure

Introduzione

Negli ultimi anni stiamo assistendo ad un crescente interesse, sia accademico che nel mercato del lavoro, per tutte quelle specializzazioni incentrate sul dato. È possibile individuare tre principali motivazioni dietro questo fenomeno:

- *IoT e social network* - decine di miliardi di dispositivi vengono connessi ad Internet ogni giorno, generando exabyte di dati. Questi ultimi non provengono solo dai social network, nei quali gli utenti condividono le proprie informazioni personali, ma anche dai sensori installati nei dispositivi di uso quotidiano, quali smartphone, smartwatch, stampanti, auto, frigoriferi, etc.
- *Democratizzazione dei dati* - cioè la diffusa disponibilità e semplicità di accesso ai dati da parte di tutti; dipendenti, segretari e manager hanno accesso a molti dati della propria azienda, dei propri clienti e dei concorrenti. Questa è essenzialmente una buona cosa, ma può anche portare alla diffusione di disinformazione e, quindi, a decisioni sbagliate, se coloro che analizzano i dati non sono dei professionisti formati a farlo.
- *Customer-centered design* - la crescita del benessere, della ricchezza e delle innovazioni tecnologiche nei paesi occidentali porta i clienti ad essere sempre più esigenti nei servizi che vengono loro offerti. Ciò spinge le aziende a trattare i propri clienti non più come numeri, bensì come persone, con particolari interessi e bisogni, il che permette di definire dei servizi personalizzati.

Dopo decenni passati ad investire e realizzare software gestionali ed applicazioni web all'avanguardia, le aziende si ritrovano in possesso di grandi quantità di dati, che spesso non sanno come utilizzare nè come gestire. Le multinazionali più importanti nel panorama IT, quali Google, Facebook ed Amazon, hanno già mostrato come dai dati degli utenti si possano ricavare informazioni molto utili, ad esempio per migliorare i servizi offerti, diminuire i costi e prevedere i trend. Il possesso di tali informazioni determina un considerevole vantaggio sulla concorrenza e, quindi, un incremento del fatturato. Uno dei settori che vede un grande numero di investimenti nell'analisi dei dati è quello finanziario; banche e compagnie di assicurazioni possono avere dei concreti vantaggi economici dalla profilazione e segmentazione dei loro clienti. Ciò ha portato alla nascita di numerose aziende specializzate nell'offrire servizi tecnologici all'avanguardia nel settore finanziario. Tali aziende vengono iden-

tificate come FinTech (Financial Technology) e risultano essere gli enti che offrono, in Italia, il maggior numero di offerte di lavoro per data scientist.

Le aziende FinTech non solo devono disporre di ingegneri informatici per la parte di analitica, ma anche di consulenti finanziari in grado di inquadrare lo specifico business case e di offrire servizi personalizzati al cliente. Il costante dialogo tra il mondo informatico e quello finanziario è alla base di ogni processo in un'azienda FinTech.

Ed è proprio nell'ambito di un progetto di data science per un'azienda FinTech di Milano che è stata concepita la presente tesi. Essa ha come oggetto di studio l'analisi di dati transazionali di un gruppo bancario.

La banca richiedeva, in particolare, il calcolo di un indicatore di rischio di concessione del credito, associato ad ogni cliente e basato sulle transazioni del suo conto corrente.

Per fare ciò, sono stati necessari la definizione ed il calcolo di numerosi KPI di finanza quantitativa, i quali hanno fornito informazioni dettagliate su diversi aspetti finanziari di ciascun cliente.

Così facendo, si è aumentato il livello di astrazione del progetto, passando dall'analisi di semplici dati transazionali sui conti correnti ad informazioni aggregate per ogni cliente.

Dall'analisi delle distribuzioni e dalla riduzione della dimensionalità dei nuovi dati, si è deciso di applicare diversi algoritmi di clustering per segmentare i clienti.

Confrontando i risultati degli algoritmi è stato possibile suddividere i clienti in tre diversi cluster; questi, dall'analisi dei corrispondenti centroidi, sono stati identificati come il cluster dei clienti a rischio, quello dei clienti in salute e quello dei clienti intermedi.

Sfortunatamente, le hard label così trovate non bastavano a soddisfare i requisiti funzionali del progetto, che richiedevano il calcolo di un indicatore numerico di rischio. Per questa ragione, si è deciso di calcolare tale indicatore finale aggregando due indicatori di rischio intermedi, uno "data driven" ed uno "rule based".

L'indicatore "data driven" è stato definito come la probabilità di appartenere al cluster di rischio elevato ed è stato calcolato eseguendo, per ogni cliente, la fase di clustering 10 000 volte, e verificando quante volte il cliente era stato etichettato come rischioso.

È stato possibile riprodurre una variabilità nell'assegnazione del cluster grazie all'utilizzo di un algoritmo non deterministico per la clustering, come il K-Means.

L'indicatore "rule based" è stato calcolato tramite somma pesata di quei KPI ritenuti comunemente essere i più rilevanti per stabilire lo stato di salute finanziaria del cliente di una banca.

La definizione e l'utilizzo di un modello statistico bayesiano hanno consentito di aggregare i due indicatori intermedi, mappati come "prior" e "likelihood", per poter ottenere l'indicatore di rischio finale, mappato come "posterior" del modello.

Infine, l'ultima parte del progetto si è incentrata sull'esecuzione di una Network Analysis, che ha consentito di individuare, a partire dagli ID dei beneficiari e degli ordinanti delle transazioni, le relazioni che ogni cliente della banca ha con i propri fornitori ed i propri clienti.

Grazie a tutte queste analisi approfondite, sono stati ottenuti degli utili indicatori da visualizzare sulle dashboard del top management del gruppo bancario.

La tesi è strutturata come di seguito specificato:

- Nel primo capitolo viene illustrata e contestualizzata la Financial Technology, sia in Italia che nel mondo. Inoltre, vengono presentate le principali tecnologie che le aziende FinTech propongono ai loro clienti.
- Nel secondo capitolo viene introdotto il progetto "RiskScore", le cui fasi vengono ripercorse nel presente elaborato. Oltre a descrivere gli obiettivi progettuali, viene brevemente presentata l'azienda FinTech presso la quale è stato svolto il progetto descritto nella presente tesi.
- Nel terzo capitolo viene analizzato nei dettagli il dataset di riferimento, controllando la qualità dei dati presenti, e viene proposta l'adozione di un differente namespace degli attributi.
- Nel quarto capitolo vengono descritti i 47 KPI progettati ed implementati dal team di lavoro. Questi sono stati molto utili per analizzare le entrate, le uscite, i saldi e l'utilizzo del conto da parte di ciascun cliente della banca.
- Nel quinto capitolo vengono presentate le fasi di data preparation e di data exploration eseguite sul dataset. In particolare, verranno analizzate le distribuzioni delle feature più rilevanti e verranno applicati due diversi algoritmi di riduzione della dimensionalità, ovvero PCA e t-SNE. I risultati di tali operazioni risulteranno essere molto utili per guidare la successiva fase di modellazione.
- Nel sesto capitolo vengono utilizzati due algoritmi di clustering, ovvero DBSCAN e K-Means, i cui risultati consentiranno di individuare tre diversi cluster di clienti. Sarà, poi, possibile associare delle caratteristiche finanziarie a ciascun cluster grazie ad un'analisi dei centroidi.
- Nel settimo capitolo viene descritta la realizzazione di due indicatori di rischio intermedio, uno data driven, quindi guidato dai risultati del clustering, ed uno rule based, cioè definito a partire da regole finanziarie.
- Nell'ottavo capitolo viene progettato un modello statistico bayesiano per poter aggregare i due indicatori intermedi in un indicatore di rischio finale. Per fare ciò, sarà necessario calcolare le distribuzioni dei due indicatori.
- Nel nono capitolo vengono presentati i risultati di una Network Analysis effettuata sulle relazioni che i clienti della banca hanno con i loro fornitori ed i loro clienti.
- Infine, nel decimo capitolo, vengono tratte delle considerazioni personali sul progetto svolto, sull'azienda ospitante e sul lavoro del data scientist in Italia.

1

La Financial Technology

Obbiettivo di questo capitolo è la trattazione della Financial Technology (FinTech); in particolare, verranno descritte le caratteristiche principali e le importanti rivoluzioni che sta apportando al mercato finanziario globale. Inoltre, verrà analizzata nei dettagli la situazione delle aziende FinTech in Italia.

1.1 Definizione

La Financial Technology, più comunemente chiamata FinTech o, tradotta in italiano, Tecnofinanza, descrive quel settore dell'innovazione che migliora i servizi finanziari attraverso le più moderne tecnologie dell'informazione e della comunicazione (ICT). All'inizio della sua storia contemporanea, la finanza ha inglobato moderne tecnologie all'interno dei propri processi, adottando, ad esempio, gli ATM negli anni '60, oppure l'online banking ed il trading negli anni '80.
Tuttavia, nel corso degli ultimi decenni, abbiamo assistito ad un'accelerazione del processo tecnologico che non ha eguali nella storia della nostra società, e ciò ha fatto si che la finanza, insieme ad altri settori quali l'agricoltura o l'istruzione, rimanessero indietro.
Per questa ragione, le attuali aziende FinTech si pongono come obbiettivo quello di dotare banche e compagnie di assicurazioni di moderni strumenti atti a migliorare i propri processi e, di conseguenza, i propri ricavi.

1.2 Tecnologie principali

Tutte le più moderne tecnologie che vengono offerte dalle aziende FinTech possono essere raggruppate in tre macro-categorie: l'Intelligenza Artificiale, la Robotic Process Automation e le blockchain.

1.2.1 Intelligenza Artificiale

Il termine Intelligenza Artificiale (IA) è spesso utilizzato in modo generico per indicare una pletora di tecnologie differenti, volte ad aumentare le conoscenze del-

l'operatore umano in un particolare dominio. Tipici task in cui si fa largo uso di algoritmi di IA sono la predizione di trend, la segmentazione della clientela ed il calcolo di indicatori per il risk management. Un'altra interessante applicazione dell'IA nel mondo FinTech è la creazione di robo-advisor.

Questi ultimi rappresentano una tipologia di entità (o, in termine tecnico, bot) che forniscono consulenze finanziarie sulla gestione di investimenti online con il minimo intervento da parte dell'uomo. Essi sfruttano algoritmi progettati da consulenti finanziari, esperti di investimenti e data scientist, e vengono codificati da ingegneri informatici all'interno di software bancari, oppure in KPI a pagamento sul web. I robo advisor utilizzano i loro algoritmi per allocare, gestire ed ottimizzare gli asset dei clienti in maniera automatica, per investimenti sia a breve che a lungo termine.

1.2.2 Robotic Process Automation

La Robotic Process Automation (RPA) afferisce a tutte quelle tecnologie coinvolte nell'automazione dei processi lavorativi umani che possono essere, invece, eseguiti in maniera automatica da un programma. Tipicamente, nelle aziende, un gran numero di attività sono ripetitive e vengono svolte da operatori umani poco qualificati, quali, ad esempio, i segretari.

Nella più vecchia visione aziendale, il segretario inserisce ed elabora manualmente pratiche ed appuntamenti, interfacciandosi con il programma gestionale dell'azienda tramite una Graphic User Interface (GUI). Nella Robotic Process Automation l'applicativo apprende i task osservando l'interazione che il segretario ha con la GUI (tramite, ad esempio, apprendimento non supervisionato, o con regole di inferenza), riuscendo, poi, ad imitarne il comportamento. La Robotic Process Automation è un processo in atto ormai da decenni.
Inizialmente utilizzato per esonerare gli operai ed i magazzinieri dei lavori più pesanti, ora viene implementato per liberare da compiti ripetitivi e noiosi dipendenti che possono, invece, essere impiegati per task mentalmente più stimolanti.

1.2.3 Blockchain

Le blockchain sono un'altra tecnologia finanziaria che sta iniziando ad essere utilizzata da banche in tutto il mondo. A differenza delle tecnologie già menzionate, esse sono state sviluppate specificatamente per le istituizioni finanziare e, quindi, presentano caratteristiche direttamente collegate al mondo finanziario; una di esse è lo smart contract, un programma che esegue automaticamente contratti tra acquirenti e venditori.

La caratteristica principale delle blockchain è, sicuramente, la decentralizzazione, la quale fa si che non sia necessario l'intervento di un ente terzo affidabile per eseguire delle transazioni. Nonostante le blockchain siano una tecnologia emergente e non siano ancora riuscite a conquistarsi la fiducia di tutte le istituzioni (soprattutto quelle pubbliche), è indubbio l'impatto che esse avranno nell'economia globale nell'immediato futuro.

1.3 Market size nel mondo

Gli investimenti mondiali nella Financial Technology sono aumentati del 2200% negli ultimi anni, passando da $930 milioni nel 2008 ad oltre i $22 miliardi del 2015.

La crescita del FinTech è un fenomeno globale. Nel 2018 si contano 1210 startup Fintech a livello globale con almeno 1 milione di dollari di finanziamento, in forte aumento (+66%) rispetto a due anni fa, capaci di raccogliere 43,7 miliardi di dollari, contro i 25,7 del 2017 (+70%). In particolare, startup cinesi, indiane e australiane crescono rispettivamente del 233%, del 184% e del 227% nella raccolta di investimenti.

1.3.1 Europa e Stati Uniti

L'industria FinTech di Londra ha avuto una rapida crescita negli ultimi anni; secondo l'ufficio del sindaco di Londra, quasi il quaranta per cento della forza lavoro londinese è impiegata, direttamente o indirettamente, in servizi FinTech.

In Europa, $1,5 miliardi sono stati investiti in aziende FinTech nel solo 2014; di questi $539 milioni sono stati investiti per aziende di Londra, $306 milioni per aziende di Amsterdam e $266 milioni per quelle di Stoccolma.

Dopo Londra, Stoccolma è la seconda città per numero di finanziamenti in Europa negli ultimi 10 anni.

Le aziende di Financial Technology negli Stati Uniti hanno ottenuto $12,4 miliardi di investimenti nel 2018, con un aumento del 43% rispetto all'anno precedente.

1.3.2 Asia

In Asia, hub tecnologici importanti nel settore finanziario sono Sydney (la quale, nel 2017, generava da sola il 9% del PIL nazionale), Hong Kong e Singapore; tuttavia, l'hub FinTech più grande dell'Asia e del mondo rimane, senza dubbio, quello cinese.

Nel 2015, il market size della Financial Technology in Cina superava gli $1,8 trilioni. Su 27 "unicorni" nel mondo (aziende FinTech che superano il miliardo di valutazione finanziaria), 8 sono aziende cinesi; queste hanno ottenuto $9,4 miliardi di finanziamenti ed hanno una valutazione combinata pari a quasi $100 miliardi (Figura 1.1).

1.4 Il FinTech in Italia

I volumi di investimento nel segmento FinTech in Italia risultano in netta crescita, sebbene il gap con i paesi europei, quali Francia, UK e Germania (seppur ridotto rispetto al passato), rimanga evidente. Il contesto italiano è caratterizzato da diverse e interessanti condizioni economiche e demografiche, che rappresentano un'opportunità per lo sviluppo e la diffusione di servizi FinTech. In particolare, a Milano hanno sede circa metà delle startup italiane, confermando il ruolo di aggregatore e promotore della crescita del mercato della città lombarda.

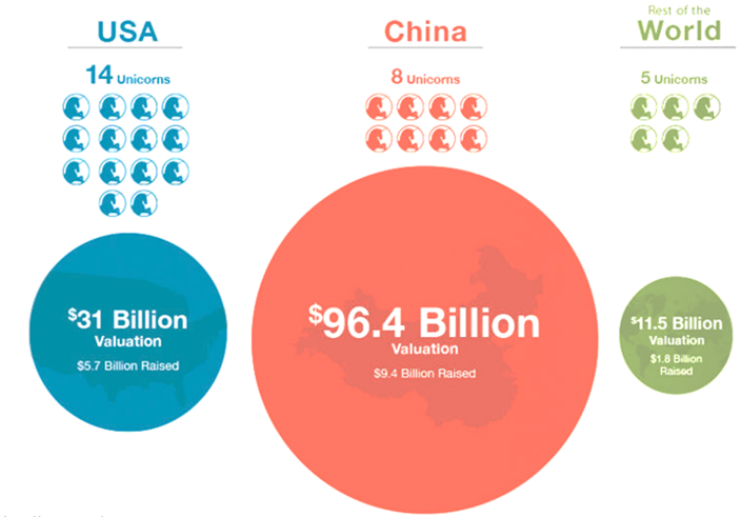

Figura 1.1. I 27 "unicorni" FinTech nel mondo

Il crescente grado di collaborazione tra gli "incumbent" (ovvero gli operatori finanziari tradizionali) e le startup digitali è un segnale molto positivo. I primi, infatti, che fino a qualche anno fa sembravano percepire le FinTech quasi come una minaccia al mantenimento delle loro quote di mercato, di recente mostrano sempre più segnali di apertura verso nuove forme di partnership strategiche o investimenti diretti in startup digitali. Tali partnership possono rivelarsi un metodo efficace per aumentare la redditività e l'efficienza dei business model.

Relativamente alla raccolta di nuovi capitali sul mercato, dall'analisi congiunta EY-Fintech District è emerso come il tema dimensionale rappresenti uno dei principali limiti allo sviluppo dell'ecosistema FinTech italiano. La maggior parte delle FinTech italiane è ancora in una fase di crescita intermedia (80% in fase Seed, Early Stage ed Early Growth), e sono rari i casi di startup che sono riuscite a concludere diversi funding round.

Nonostante un valore assoluto degli investimenti a livello italiano ancora limitato, nel 2018 e nel 2019 si è assistito a una crescita importante dei volumi; nel 2018 i fondi raccolti sono quadruplicati rispetto all'anno precedente, passando da 54 a 200 milioni di euro, per toccare quota 261 milioni nel 2019, registrando un CAGR (ovvero il tasso annuo di crescita composto) pari al +62%.

Altri ostacoli alla diffusione del FinTech in Italia sono la presenza di vecchi sistemi legacy nelle banche e le lente procedure burocratiche delle istituzioni.

2

Il progetto “RiskScore”

In questo capitolo viene presentato il progetto di tesi. Viene, innanzitutto, introdotta l’azienda in cui si è svolto il progetto, successivamente, viene descritta la problematica che il cliente ha richiesto di risolvere. Infine, vengono elencati i vari task che è stato necessario effettuare per poter completare il progetto.

2.1 L’azienda

L’azienda che si è offerta di ospitare il tirocinante è la Virtual B, un’azienda FinTech di Milano.

Essa è stata fondata nel 2010 da un team di esperti con un forte background nel settore finanziario. Nel 2011 la società ha lanciato AdviseOnly, il primo robo-advisor in Europa. Oggi, Virtual B fornisce analisi e soluzioni digitali per la gestione patrimoniale con una forte attenzione all’innovazione e alle tecnologie rivoluzionarie. L’azienda è di piccole dimensioni (10-15 dipendenti) con figure professionali molto specializzate nei settori finanza ed IT.
I clienti della Virtual B sono, generalmente, banche e compagnie di assicurazioni che richiedono consulenze finanziarie o analisi dei propri dati interni. Per quanto riguarda quest’ultimo servizio, la Virtual B si occupa, in genere, delle fasi a monte della pipeline, lasciando ad aziende esterne il compito di realizzare le dashboard per meglio visualizzare i risultati delle analisi.

2.2 Obbiettivo del progetto

L’obbiettivo del progetto “RiskScore” è quello di analizzare dei dati transazionali per un’importante banca del Nord Italia. In particolare, è stato richiesto dai manager della banca il calcolo di un indicatore di rischio per poter meglio valutare se concedere o meno un prestito ai clienti. È stata concessa piena libertà alla Virtual B sulla scelta di come calcolare tale indicatore. Come supporto alle analisi, è stato fornito un dataset di dati transazionali riportanti i movimenti in entrata ed in uscita nei conti correnti di un campione di clienti della banca. Nell’ambito delle attività

connesse alla presente tesi sono state curate tutte le fasi di analisi del progetto, in collaborazione con gli altri data scientist dell’azienda. Come tool sono stati utilizzati numerosi moduli Python, tra cui i noti `pandas` e `scikit-learn`.

2.3 Lista dei task

Per poter meglio organizzare le analisi, si è deciso di suddividere il progetto in una sequenza di fasi di lavoro o task. Al termine di ogni fase, i risultati intermedi sono stati periodicamente esposti e discussi sia con il responsabile della data science, sia con quello dell’IT dell’azienda.
Le fasi principali inizialmente pianificate sono state le seguenti:

1. realizzazione di KPI di finanza quantitativa;
2. data preparation;
3. data exploration;
4. modelling;
5. realizzazione di indicatori di rischio.

Esse seguono il famoso open standard CRISP-DM (Cross-Industry Standard Process for Data Mining, Figura 2.1). L’unica principale differenza che spesso si nota tra i progetti che adottano questo standard è l’ordine delle fasi di data preparation e data exploration, che in certi casi, come nel nostro, può risultare invertito. Il motivo di questa scelta è legato al dataset in esame e al significato che si dà alla data exploration.
Nel nostro dataset, trattandosi di dati transazionali, la data exploration si limiterebbe a contare il numero di dati mancanti e ad individuare righe duplicate. Lo studio della forma delle diverse distribuzioni e dello spazio delle feature è possibile solo dopo aver aggregato i dati per cliente, ed aver applicato i KPI di finanza quantitativa. Per questo motivo, nella presente tesi, è stato deciso di esporre dapprima la data preparation, e, successivamente, la data exploration.

Spesso i processi di analisi dei dati (così come qualsiasi progetto di ingegneria del software) necessitano di modifiche ed aggiunte in itinere. Questo perchè, durante lo sviluppo del progetto, vengono aggiunti dal cliente nuovi requisiti, oppure perchè nascono nuove idee progettuali in base ai risultati intermedi ottenuti.

A progetto avviato, il team di lavoro ha deciso di aggiungere ai task precedentemente concordati anche i seguenti:

6. creazione di un modello bayesiano;
7. Network Analysis.

Il modello bayesiano è stato necessario per poter aggregare in maniera statisticamente formale due indicatori di natura diversa, mentre la Network Analysis è stata utilizzata per sfruttare l’informazione sui legami che ogni cliente della banca ha con i propri fornitori ed i propri clienti.

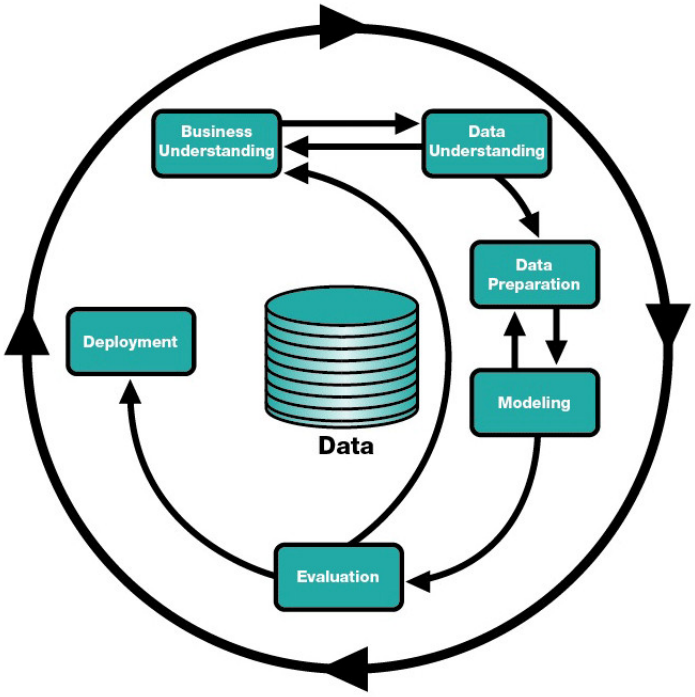

Figura 2.1. Process diagram dello standard CRISP-DM

3

Descrizione del dataset di riferimento

In questo capitolo descriveremo il dataset fornitoci per eseguire le analisi del progetto. Oltre ad elencare il significato delle varie colonne, verrà discussa l'importante scelta del namespace da adottare durante le fasi di analisi.

3.1 Dati raw

Il dataset fornitoci dalla banca è un singolo file Excel con 32 581 righe e 19 colonne. Ogni riga corrisponde ad una transazione di un particolare cliente. Le transazioni possono essere in ingresso o in uscita sul conto corrente dei clienti. Sono state memorizzate le transazioni avvenute nel corso di 12 mesi, dall'1 gennaio 2018 al 31 dicembre 2018, su un campione di 51 clienti. È importante sottolineare il fatto che i clienti in questione non sono degli individui, bensì delle aziende; perciò, sono memorizzate nel dataset anche informazioni relative all'ultimo bilancio da loro redatto. Di seguito viene riportato un elenco delle feature (colonne) del dataset, accompagnate da una breve descrizione e dal tipo dei loro valori (Figura 3.1).

Innazitutto, è possibile notare come, all'interno del medesimo dataset, siano contenuti attributi appartenenti ad entità differenti. In particolare:

- il codice NDG, il codice fiscale, il fatturato, la natura giuridica, il valore della produzione, lo schema del bilancio, la descrizione dello schema del bilancio e la data del bilancio fanno riferimento all'entità *cliente*;
- la tipologia di pagamento, il numero della transazione, la data della transazione e l'importo in euro fanno riferimento all'entità *transazione*.

Ci sono, inoltre, due coppie di attributi che fanno riferimento all'entità *beneficiario* e all'entità *ordinante*.
Tuttavia, esse non sono entità indipendenti come *cliente* e *transazione* in quanto, in base alla direzione della dispozione, la transazione può essere diretta verso o dal conto corrente del cliente; perciò, nel primo caso il beneficiario coinciderà con il cliente, mentre nel secondo caso sarà l'ordinante ad essere il cliente. Quindi ogni riga del dataset descrive una transazione avvenuta tra un cliente della banca e un proprio fornitore o un proprio cliente.

Attributo	Descrizione	Tipo
id_riga	ID della riga nel file Excel.	int
data_riferimento	Data dello snapshot dei dati.	date
tipologia_pagamento	Tipo di pagamento effettuato.	string
numero_disposizione	ID della transazione.	int
direzione_disposizione	Indica se la transazione è un versamento o un prelievo.	char
data_regolamento	Data della transazione.	date
importo_euro	Ammontare in euro della transazione.	float
iban_ordinante	IBAN di colui che ha ordinato la transazione.	string
anagrafica_ordinante	Codice anagrafico dell'ordinante.	string
iban_beneficiario	IBAN di colui che ha ricevuto la transazione.	string
anagrafica_beneficiario	Codice anagrafico del beneficiario.	string
ndg	ID interno del cliente della banca.	int
natura_giuridica	Natura giuridica dell'azienda cliente della banca.	string
codice_fiscale	Codice fiscale dell'azienda cliente della banca.	string
fatturato	Il fatturato dell'azienda cliente della banca.	float
valore_produzione	Il valore della produzione dell'azienda cliente della banca.	float
data_bilancio	Data dell'ultimo bilancio redatto.	date
schemaricl	Schema dell'ultimo bilancio redatto (numerico).	int
descrizione_schemaricl	Schema dell'ultimo bilancio redatto (stringa).	string

Figura 3.1. Gli attributi del dataset di riferimento

Nonostante l'inserimento di attributi di entità differenti all'interno dello stesso dataset sia un'abitudine da evitare (soprattutto a causa di problemi relativi all'aggiornamento o allo spreco di spazio), essa è sfortunatamente una pratica molto diffusa, soprattutto in banche e grandi multinazionali che si affidano ad aziende di consulenza informatica esterne. Le aziende di consulenza spesso prediligono la velocità di progettazione alla qualità delle architetture software da loro realizzate, producendo dei risultati che, come in questo caso, sono discutibili.

Un altro segno di cattiva progettazione del dataset sono i due attributi sullo schema del bilancio; essi codificano esattamente la stessa informazione, ma in modo diverso (uno come numero, l'altro come stringa).

Da notare, infine, la totale inutilità dell'attributo sulla data dello snapshot, che codifica un'informazione che non deve essere associata alle singole istanze, bensì all'intero dataset.

3.1.1 Osservazioni sulla qualità dei dati

Appena ricevuto il dataset si è effettuata una semplice verifica della qualità dei dati memorizzati nelle varie colonne. Per velocizzare il lavoro, è stato utilizzato il tool di reportistica `ProfileReport`, contenuto nel modulo di Python `pandas_profiling` .

In Figura 3.2 vediamo come il dataset riporta 6854 celle mancanti, pari circa all'1% delle celle totali. Questa è un'informazione piuttosto rassicurante, soprattutto se le poche celle mancanti appartengono ad attributi secondari. L'assenza di righe duplicate riduce le possibilità che il dataset sia stato manomesso, garantendo la bontà delle informazioni presenti.

Dataset statistics

Number of variables	19
Number of observations	32580
Missing cells	6854
Missing cells (%)	1.1%
Duplicate rows	0
Duplicate rows (%)	0.0%
Total size in memory	4.7 MiB
Average record size in memory	152.0 B

Figura 3.2. Statistiche sul dataset generate da `pandas_profiling`

Da un'analisi più approfondita è stato possibile scoprire che la maggior parte delle celle mancanti sono relative alle colonne sugli IBAN dell'ordinante e del beneficiario della transazione, informazioni relativamente secondarie.

Inoltre, gli attributi sull'ID della riga e sul numero della disposizione riportano entrambi valori unici: essi sono, quindi, equivalentemente utilizzabili come chiave primaria del dataset. Come è stato già accennato, la data del riferimento risulta essere un valore costante per tutte le transazioni, essendo essa semplicemente la data in cui è stato realizzato lo snapshot del dataset. Infine, l'attributo per l'anagrafica dell'ordinante risulta essere di un tipo non supportato dal tool di analisi, probabilmente a causa di qualche suo valore non codificato correttamente.

3.2 Namespace adottato

Su richiesta del responsabile del reparto di data science dell'azienda, i nomi degli attributi sono stati sostituiti da un namespace in lingua inglese, con dei termini più intuitivi e coerenti tra loro. Per la creazione del namespace si è deciso di utilizzare il pattern "entità_attributo" per due ragioni:

- riflette una corretta traduzione inglese, rispettando, quindi, il pattern grammaticale aggettivo_nome;
- gli attributi vengono implicitamente raggruppati per nome di entità, rendendo più facile capire a quale entità ciascun attributo faccia riferimento.

Per la traduzione inglese è stato seguito lo standard della terminologia finanziaria adottata dalla European Central Bank (Figura 3.3).

Dall'analisi della figura si può notare che sono stati eliminati i seguenti attributi:

- *id_riga*, in quanto è preferibile utilizzare come chiave della transazione l'attributo univoco *numero_disposizione*, significativamente più rilevante nella terminologia bancaria;
- *data_riferimento*, in quanto inutile, perchè indica semplicemente la data dello snapshot dell'elenco delle transazioni;

Originale	Traduzione
tipologia_pagamento	transaction_payment_type
numero_disposizione	transaction_id
data_regolamento	transaction_date
importo_euro	transaction_amount
iban_ordinante	sender_iban
anagrafica_ordinante	sender_registration_id
iban_beneficiario	receiver_iban
anagrafica_beneficiario	receiver_registration_id
ndg	client_id
natura_giuridica	client_company_type
codice_fiscale	client_fiscal_code
fatturato	client_revenues
valore_produzione	client_production_value
data_bilancio	client_balance_date
descrizione_schemaricl	client_balance_schema_description

Figura 3.3. Namespace adottato per le analisi

- *direzione_disposizione*, in quanto l'informazione sulla direzione della transazione (da o verso il conto corrente del cliente) può essere resa implicita aggiungendo il segno su *importo_euro*;
- *schemaricl*, perchè la stessa informazione è codificata nell'attributo *descrizione_schemaricl*.

4

Definizione di KPI di finanza quantitativa

Nel capitolo viene descritta la progettazione di una serie di KPI di finanza quantitativa, utili per estrarre informazioni finanziarie a partire dalle transazioni dei clienti. Affinchè tali funzioni siano facilmente riutilizzabili in futuro su altri dataset e da altri dipendenti dell'azienda, viene presentata la realizzazione di una interfaccia per l'accesso ai KPI.

4.1 Progettazione dei KPI

La finanza quantitativa è l'applicazione di matematica, statistica e informatica alla risoluzione di problemi finanziari. È stato necessario progettare e realizzare dei KPI di finanza quantitativa che permettessero di calcolare utili informazioni finanziarie dei clienti a partire dalle loro transazioni. La loro progettazione ha richiesto una costante interazione con i consulenti finanziari dell'azienda, soprattutto per poter comprendere la terminologia tecnica e scegliere gli indicatori più adatti per questo progetto. I KPI implementati sono stati 37, suddivisi in 4 categorie, ognuna utile a fornire informazioni differenti su ciascun cliente. Essi, quindi, vengono calcolati su tutte quelle transazioni (in ingresso o in uscita) che fanno riferimento al medesimo conto corrente (e, quindi, allo stesso cliente della banca).

4.1.1 Analisi delle entrate

I KPI per l'analisi delle entrate sono nove indicatori calcolati sulle transazioni in ingresso ai conti correnti dei clienti. Essi sono:

1. *monthly revenues* - somma dei flussi in ingresso, divisi per mese;
2. *median monthly revenues* - mediana dei monthly revenues;
3. *IQ monthly revenues* - range interquantilico dei monthly revenues;
4. *trend of revenues* - coefficiente della regressione lineare calcolata sui monthly revenues;
5. *recent revenues* - media dei monthly revenues degli ultimi 3 mesi;
6. *monthly number of inflows* - numero delle transazioni in ingresso, divise per mese;

7. *max revenues* - valore massimo tra i monthly revenues;
8. *distribution of revenues* - lista di tutti gli ordinanti delle transazioni in ingresso (quindi di tutti i clienti dei clienti della banca) associati alla frequenza di revenues relativa all'ammontare di tutti gli ingressi per lo stesso beneficiario;
9. *revenues concentration index* - entropia calcolata su distribution of revenues.

4.1.2 Analisi delle uscite

I KPI per l'analisi delle uscite sono nove indicatori calcolati sulle transazioni in uscita dai conti correnti dei clienti. Essi sono:

10. *monthly expenses* - somma dei flussi in uscita, divisi per mese;
11. *median monthly expenses* - mediana dei monthly expenses;
12. *IQ monthly expenses* - range interquantilico dei monthly expenses;
13. *trend of expenses* - coefficiente della regressione lineare calcolata sui monthly expenses;
14. *recent expenses* - media dei monthly expenses degli ultimi 3 mesi;
15. *monthly number of outflows* - numero delle transazioni in uscita, divise per mese;
16. *max expenses* - valore massimo tra le monthly expenses;
17. *distribution of expenses* - lista di tutti i beneficiari delle transazioni in uscita (quindi di tutti i fornitori dei clienti della banca) associati alla frequenza di expenses relativa all'ammontare di tutte le uscite dello stesso ordinante;
18. *expenses concentration index* - entropia di distribution of expenses.

Come è possibile notare, essi sono gli stessi KPI per l'analisi delle entrate, calcolati, però, sulle transazioni in uscita, anzichè sulle transazioni in ingresso.

4.1.3 Analisi dei saldi

I KPI per l'analisi dei saldi sono tredici indicatori calcolati sommando l'ammontare in euro delle transazioni in ingresso e in uscita in un determinato periodo, valutando, quindi, se il saldo sia positivo o negativo. Essi sono:

19. *monthly number of days with a negative balance* - somma dei giorni con saldo negativo, mese per mese;
20. *total number of days with a negative balance* - somma dei giorni con saldo negativo nell'intero periodo;
21. *monthly number of days with a positive balance* - somma dei giorni con saldo positivo, mese per mese;
22. *total number of days with a positive balance* - somma dei giorni con saldo positivo nell'intero periodo;
23. *monthly ratio negative/positive* - rapporto tra il numero di giorni con saldo negativo e quelli con saldo positivo, mese per mese;
24. *total ratio negative/positive* - rapporto tra il numero di giorni con saldo negativo e quelli con saldo positivo nell'intero periodo;
25. *monthly balance* - saldo di fine mese, mese per mese;
26. *median monthly balance* - mediana del monthly balance;
27. *IQ monthly balance* - range interquantilico del monthly balance;

28. *trend of balance* - coefficiente di regressione lineare calcolato sui monthly balance;
29. *min balance* - minimo di tutti i saldi giornalieri;
30. *max balance* - massimo di tutti i saldi giornalieri;
31. *monthly ratio outflows/inflows* - rapporto tra il numero di transazioni in uscita e il numero di transazioni in entrata, mese per mese.

4.1.4 Analisi dell'uso del conto

I KPI per l'analisi dell'uso del conto sono sei indicatori calcolati valutando il numero di transazioni avvenute sul conto in un determinato periodo.

32. *number of transactions* - numero totale di transazioni nell'intero periodo;
33. *monthly number of transactions* - numero di transazioni mensili, mese per mese;
34. *recent to historical negative ratio* - media del rapporto tra il numero di uscite ed il numero di entrate mensili negli ultimi 3 mesi, meno la media del rapporto tra il numero di uscite ed il numero di entrate mensili nei restanti mesi;
35. *recent to historical usage ratio* - media del numero delle transazioni negli ultimi 3 mesi meno la media del numero delle transazioni nei restanti mesi;
36. *unusual recent account negative activity* - se il min balance è avvenuto negli ultimi 3 mesi, allora true, altrimenti false;
37. *usual recent account positive activity* - se il max balance è avvenuto negli ultimi 3 mesi, allora true, altrimenti false.

4.2 Realizzazione di un'interfaccia per i KPI

La corretta realizzazione di tutti i KPI ha, ovviamente, richiesto tempo ed energie; per questa ragione si è voluto inserirli in un modulo Python accessibile in maniera facile ed intuitiva attraverso un'interfaccia. In questo modo, le funzioni per i KPI potranno essere riutilizzate in futuro su altri dataset e da altri dipendenti dell'azienda, semplicemente fornendo parametri differenti alle chiamate dei KPI esposti dall'interfaccia.

Essendo gli indicatori classificabili in quattro categorie differenti, si è pensato di inserire un ulteriore livello di astrazione tra gli script del calcolo dei KPI e l'interfaccia che li espone. Tale strato è costituito dalle seguenti quattro classi Python:

- *InflowClass* - classe per i KPI dell'analisi delle entrate;
- *OutflowClass* - classe per i KPI dell'analisi delle uscite;
- *BalanceClass* - classe per i KPI dell'analisi dei saldi;
- *AccountClass* - classe per i KPI dell'analisi dell'uso del conto.

Le funzioni per il calcolo dei KPI vengono quindi chiamate direttamente dalle quattro classi di analisi, le quali vengono istanziate dall'interfaccia finale per esporre i loro metodi all'utente.

Il diagramma UML dell'architettura stratificata appena descritta è visibile in Figura 4.1.

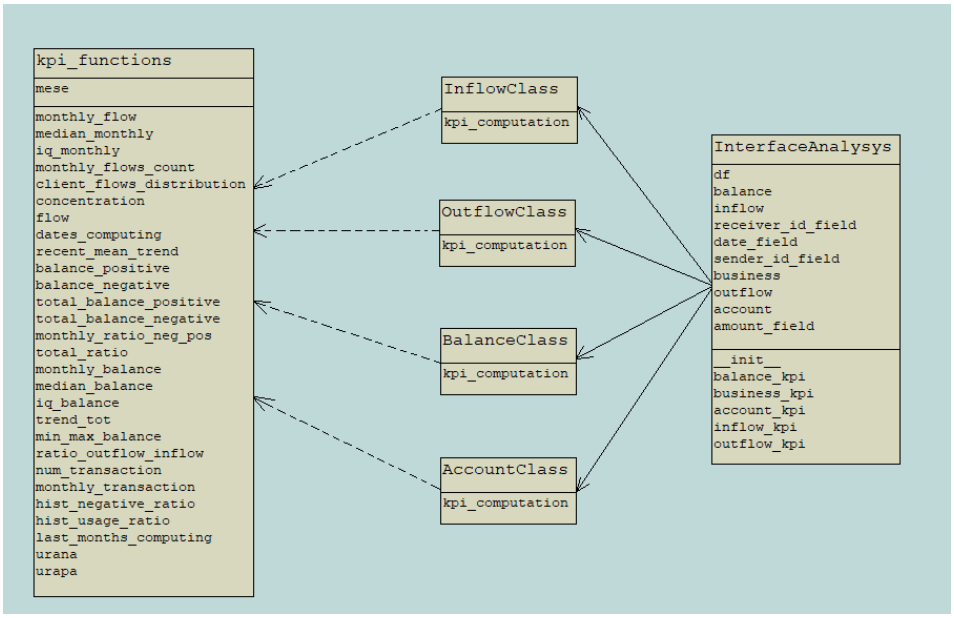

Figura 4.1. Class diagram dei KPI

5

Attività di Data Preparation e Data Exploration

In questo capitolo vengono descritte le attività di Data Preparation e di Data Exploration che sono state effettuate dal team di lavoro. Esse verranno illustrate in dettaglio, indicando non solo le varie scelte tecniche che sono state prese, ma anche le motivazioni dietro di esse. I risultati ottenuti da queste fasi iniziali sono state di rilevante importanza per guidare le successiva fase di modellazione.

5.1 Calcolo KPI di ogni cliente

I dati transazionali sono stati raggruppati per cliente, eseguendo su ciascun gruppo le funzioni per il calcolo dei KPI. I gruppi, quindi i clienti totali, sono risultati essere 51. Il calcolo dei KPI è stato molto semplice e privo di problemi, soprattutto grazie alla pratica architettura precedentemente realizzata. I dati mancanti appartenevano principalmente a colonne non rilevanti per il calcolo degli indicatori; quei pochi valori nulli in colonne rilevanti, invece, sono stati semplicemente sostituiti dalla media dei valori di quella colonna. Con i valori dei KPI per ciascun cliente è stato, quindi, realizzato un nuovo dataset, avente 51 righe (una per ogni cliente) e 191 colonne (una per ogni KPI). Il numero delle colonne potrebbe risultare spropositato, ma ricordiamo che molti dei KPI descritti nel capitolo precedente erano mensili, quindi, per ciascuno di essi, sono associate 12 colonne differenti. Inoltre, si è deciso di conservare anche tutti gli attributi delle transazioni che erano associate al cliente, come il fatturato o la natura giuridica dell'azienda.

5.1.1 Curse of dimensionality

Dataset costituiti da un grande numero di attributi e da un relativo piccolo numero di sample rischiano di soffrire della "curse of dimensionality", che rende più difficili le analisi ed il training di modelli di machine learning. Esistono due possibili soluzioni per aggirare questo problema:

1. aumentare il numero dei sample;
2. ridurre il numero delle feature.

Si è tentato di richiedere alla banca i dati di altri clienti da aggiungere a quelli del dataset, ma tale possibilità ci è stata negata.
È stata, inoltre, vagliata l'opzione di eseguire una Data Augmentation, per aumentare artificialmente il numero di sample, ma tale possibilità è stata scartata a causa dell'elevato rischio di introdurre rumore nei dati. Per quanto riguarda, invece, la riduzione del numero di feature, essa è un'opzione valida solo se vengono individuate feature scarsamente rilevanti. Il team di lavoro ha deciso di procedere utilizzando tutte le feature, passando alla parte di modellazione solo dopo aver effettuato un'attenta esplorazione visiva dei dati, studiando le distribuzioni ed utilizzando tecniche di riduzione della dimensionalità.

5.2 Analisi delle distribuzioni

Una volta ottenuto il dataset con i valori aggregati per ciascun cliente, si è passati all'analisi delle distribuzioni dei valori delle feature. Ovviamente, essendoci quasi duecento colonne, si è deciso di analizzare solo le distribuzioni delle feature ritenute più interessanti.

5.2.1 Fatturato

Il fatturato di un'azienda è sicuramente una caratteristica rilevante per la segmentazione della clientela, in quanto è una misura direttamente collegata alle dimensioni dell'azienda. In Figura 5.1 è mostrato l'istogramma della distribuzione dei valori della feature *client_revenues*. Tale distribuzione è di tipo power law, in quanto i valori sono enormemente sbilanciati verso una parte del grafico. In particolare, nel nostro dataset, la clientela della banca è costituita principalmente da piccole o medie imprese, quindi con un fatturato ridotto; vi è, poi, un piccolo numero di aziende con un fatturato molto più grande. Il valore dell'estremo inferiore è di 8000 euro, mentre quello dell'estremo superiore è di olre 3 miliardi e mezzo di euro.

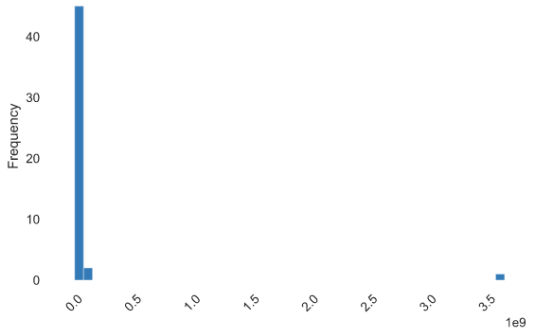

Figura 5.1. Istogramma di *client_revenues*

In una distribuzione di questo tipo non ha senso calcolare metriche quali media e deviazione standard; invece, è stata calcolata la skewness, che ci fornisce un'indi-

cazione di quanto la distribuzione sia asimmetrica rispetto ad una gaussiana, che ha una skewness di riferimento pari a 0. La skewness di *client_revenues* è di quasi 7, confermando l'elevata asimmetria che contraddistingue una distribuzione power law.

5.2.2 Numero di transazioni

Il numero di transazioni totali effettuate ci permette di capire quanto un'azienda sia attiva sul suo conto corrente; tale parametro si ottiene, quindi, sommando il numero di ingressi e di uscite nell'intero periodo. In Figura 5.2 vediamo una situazione simile a quella di *client_revenues*, in cui la maggior parte delle aziende ha effettuato un numero limitato di transazioni, mentre poche aziende ne hanno effettuate moltissime. I valori estremi sono pari a 4 e a 7675 transazioni, mentre la skewness è di 4,6.

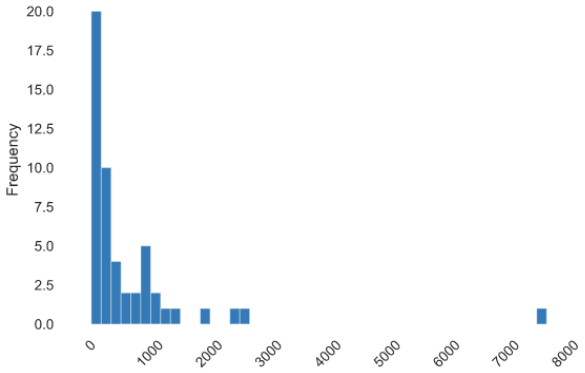

Figura 5.2. Istogramma di *num_transactions*

5.2.3 Natura giuridica

La natura giuridica dell'impresa è il modello organizzativo, amministrativo, fiscale e contabile con cui viene condotta un'azienda, secondo le norme del Codice Civile. Nell'istogramma di Figura 5.3 vediamo come la distribuzione delle varie forme giuridiche all'interno del campione segua quello delle aziende del Nord Italia, ad indicazione del fatto che il campione è rappresentativo soprattutto di quest'area geografica. Infatti, le aziende del Nord sono principalmente SRL, ma con una buona rappresentanza anche di SPA (che, invece, sono quasi assenti nel Sud).

5.2.4 Trend del saldo

La Figura 5.4 mostra la distribuzione dei valori di *trend_balance*, cioè del coefficiente della regressione lineare calcolata su tutti i saldi mensili, per ogni cliente. Un valore positivo indica che il cliente tende ad avere più denaro in ingresso che in uscita

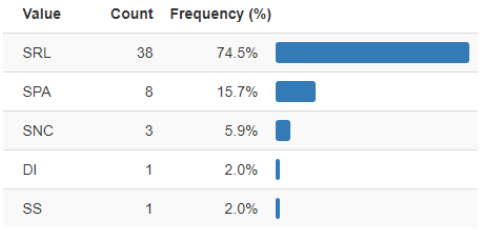

Figura 5.3. Istogramma di *client_company_type*

dal suo conto, mentre un valore negativo indica il contrario. Questa distribuzione è molto più simmetrica rispetto alle altre, avendo un valore di skewness pari a 2. Agli estremi della distribuzione sono visibili degli outlier, che indicano la presenza di aziende che hanno avuto molte più uscite che ingressi, ed aziende, invece, che hanno avuto molti più ingressi che uscite.

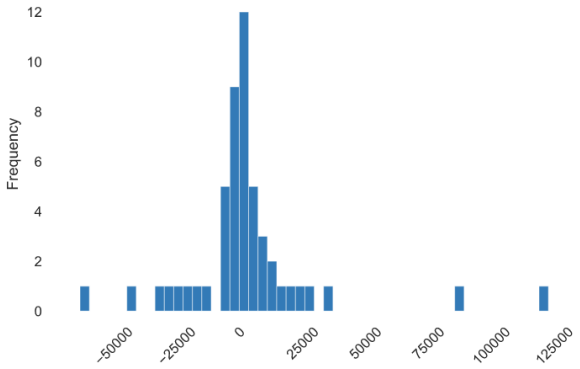

Figura 5.4. Istogramma di *trend_balance*

5.3 Normalizzazione

Prima di realizzare dei modelli di analisi, è spesso molto utile applicare un feature scaling. Poiché i range dei valori variano molto tra le varie feature, molti algoritmi di machine learning non funzionerebbero correttamente con i dati in nostro possesso. Per esempio, molti algoritmi calcolano la distanza euclidea per misurare la distanza tra due data point. Se una delle feature ha un range di valori molto più ampio delle altre (quindi, ha una varianza maggiore), la distanza tra i due data point sarebbe determinata principalmente da essa. Con un feature scaling, invece, ogni feature contribuirebbe proporzionalmente al calcolo della distanza finale. Una delle tecniche di feature scaling più utilizzata è la normalizzazione, che scala i valori in modo che rientrino nel range tra 0 ed 1.

C'è, però, un problema: dalle analisi sulle distribuzioni delle feature del dataset è risultato che i sample rispettano il principio di Pareto, in cui, cioè, il 20% degli individui produce l'80% del valore. In particolare, il 20% delle aziende con più uscite contribuisono all'80% delle uscite totali dell'intero campione. Vale lo stesso discorso per le entrate. La grande asimmetria delle distribuzioni di molte feature farebbe si che, durante la loro normalizzazione, i pochi valori più grandi diventerebbero pari ad 1, mentre i molti valori piccoli verrebbero schiacciati sullo 0. Per evitare questa normalizzazione anomala, il team di lavoro ha prima calcolato la funzione logaritmo naturale sui valori di tutte quelle feature che riportavano una skewness troppo elevata, e, quindi, che erano estremamente asimmetriche. Infine, l'applicazione della normalizzazione sui risultati della funzione logaritmo naturale ha consentito di ottenere dei valori più uniformemente distribuiti tra 0 ed 1.

5.4 Riduzione della dimensionalità

La riduzione della dimensionalità è una trasformazione dei dati da uno spazio multidimensionale ad uno con un minor numero di dimensioni, mantenendo le proprietà più rilevanti che avevano i dati nello spazio originario. Avere un dataset con molte feature può, da una parte, arricchire l'analisi di molte informazioni interessanti; dall'altra, però, può causare numerosi problemi, tra cui:

- l'aumento della varianza nei dati, la quale può causare overfitting, soprattutto se si ha un numero di sample minore al numero di feature (curse of dimensionality);
- la densità e la distanza tra i dati che perdono di significato;
- un'esplosione combinatoria, che rende l'esecuzione del processo di calcolo computazionalmente intrattabile.

Il team di lavoro ha utilizzato la riduzione della dimensionalità soltanto a scopo esplorativo, quindi per visualizzare i data point in uno spazio bidimensionale o tridimensionale, ottenendo informazioni utili per guidare, poi, la successiva fase di modellazione.

Esistono numerosi algoritmi per la riduzione della dimensionalità; si è deciso di utilizzarne due, ovvero PCA e t-SNE, per poi confrontare i risultati.

5.4.1 PCA

La Principal Component Analysis, più comunemente conosciuta come PCA, è indubbiamente la tecnica più utilizzata per la riduzione della dimensionalità. Essa ha il vantaggio di fornire informazioni quantitative sul peso che ha ogni feature nella descrizione della varianza complessiva del dataset. Per questo motivo, la PCA è molto utile ad individuare le feature più rilevanti, eliminando quindi quelle che generano semplicemente rumore nel dataset.

Il suo funzionamento è relativamente semplice; essa individua la direzione di massima varianza dei dati, definisce lungo quella direzione un autovettore, chiamato "prima componente principale", e costruisce, poi, le altre componenti principali ortogonalmente tra loro.

Proiettando i data point su ogni singola componente principale, costruisce, quindi, uno spazio di dimensionalità ridotta.

Per sua natura, la PCA non richiede una particolare attenzione nel setting degli iperparametri, dal momento che i valori di default riportati nella documentazione di `scikit-learn` vanno bene per la maggior parte dei dataset. È, però, importante fare attenzione al numero delle componenti principali scelte; infatti, maggiore è tale numero, e maggiore è la capacità del modello di spiegare la varianza totale dei dati iniziali. Tuttavia, aumentare i parametri di un modello causa anche un incremento della sua complessità. Per i fini della sola Data Exploration, risulta obbligata la scelta di due o al massimo tre componenti principali, in modo da poter rappresentare i dati in uno spazio 2D o 3D.

Nel modello della PCA realizzato con 3 componenti principali, gli autovalori ottenuti sono stati 3,06, 1,17 e 0,89, con una varianza spiegata, rispettivamente, del 35%, del 13% e del 10%, per un totale di quasi il 60% di varianza spiegata dal modello. Nelle Figure 5.5, 5.6 e 5.7 sono riportate le 10 feature più determinanti per ciascuna componente principale, associate al loro peso, chiamato anche "load factor".

Feature	Load factor
iq_monthly_revenues.9	0,147
iq_monhly_revenues.5	0,14
iq_monthly_revenues.6	0,138
iq_monthly_revenues.2	0,135
monthly_negative_days.6	0,13
monthly_negative_days.9	0,127
monthly_expenses.4	-0,126
iq_monthy_revenues.10	0,125
total_negative_days	0,122
monthly_negative_days.4	0,12

Figura 5.5. Feature e load factor della PC1

Feature	Load factor
monthly_ratio_neg_pos.6	0,163
monthly_negative_days.7	0,15
monthly_positive_days.8	-0,149
monthly_positive_days.12	-0,143
monthly_positive_days.10	-0,14
monthly_positive_days.9	-0,139
monthly_positive_days.3	-0,139
monthly_negative_days.12	0,137
monthly_positive_days.11	-0,136
monthly_positive_days.1	-0,136

Figura 5.6. Feature e load factor della PC2

Da tali informazioni è possibile trarre le seguenti osservazioni:

Feature	Load factor
iq_monthly_expenses.1	-0,15
iq_monthly_expenses.9	-0,144
iq_monthy_revenues.8	-0,143
monthly_revenues.6	0,14
unusual_negative_activity	0,14
iq_monthly_expenses.6	-0,136
iq_monthly_expenses.5	-0,135
iq_monthly_expenses.4	-0,133
monthly_balance.11	0,13
monthly_revenues.9	0,129

Figura 5.7. Feature e load factor della PC3

1. La percentuale di varianza complessiva spiegata dalle tre componenti principali non è molto alta, ma è, comunque, sufficiente a permetterci di trarre informazioni interessanti per la Data Exploration.
2. La prima componente principale ha un autovalore nettamente superiore rispetto a quello delle altre due, indicando che lungo la sua direzione si ha una varianza maggiore rispetto alle altre direzioni.
3. La PC1 è influenzata positivamente dai ricavi mensili dei clienti, e negativamente dalle spese.
4. La PC2 è influenzata negativamente dal numero dei giorni con saldo positivo, e positivamente dal numero dei giorni con saldo negativo.
5. La PC3 è influenzata negativamente dalle spese mensili, e positivamente da ricavi mensili.

Gli esperti finanziari dell'azienda hanno sostenuto la coerenza delle osservazioni 3, 4 e 5, confermando la validità dei risultati ottenuti dalla PCA.

Sono stati poi rappresentati i risultati della PCA, sia in 2D che in 3D, associando colori diversi ai punti in base alla natura giuridica dell'azienda (l'unico attributo categorico rilevante). I grafici ottenuti sono visibili nelle Figure 5.8 e 5.9.

Osservando tali grafici è possibile concludere che:

1. Le rappresentazioni 2D e 3D sono coerenti tra loro.
2. PC1 e PC2 mostrano chiaramente le direzioni di maggiore varianza, dalle quali è possibile distinguere due cluster, uno più denso, con valori di PC1 più piccoli, ed uno meno denso, con valori di PC1 più grandi.
3. PC2 descrive molto meglio il secondo cluster che il primo.
4. PC3 sembra descrivere unicamente degli outlier dei due cluster principali, o comunque delle relazioni locali tra i due cluster.
5. *Client_company_type* non sembra essere un attributo rilevante nella suddivisione dei cluster.

5.4.2 t-SNE

Contrariamente alla PCA, che semplicemente cerca di massimizzare la varianza delle feature, t-SNE crea una spazio di dimensionalità ridotta dove sample simili sono

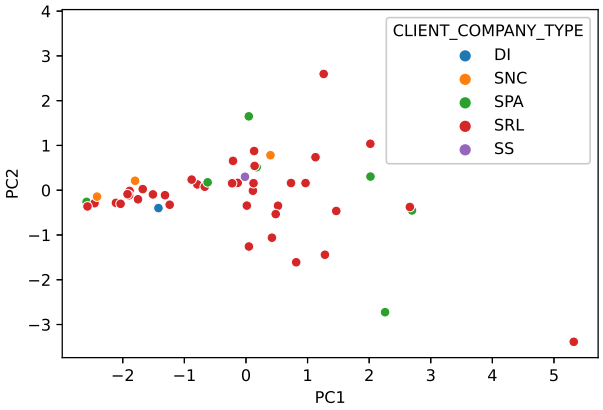

Figura 5.8. Plot 2D della PCA

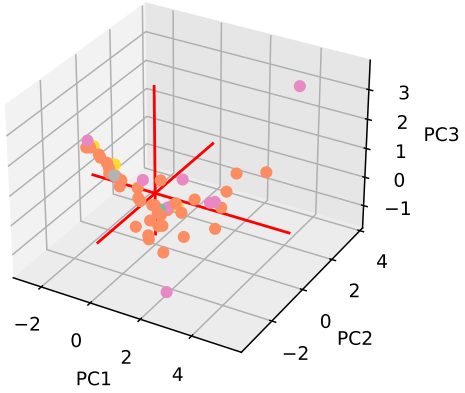

Figura 5.9. Plot 3D della PCA

modellati come punti vicini, mentre sample diversi sono modellati come punti lontani. Ciò significa che, anzichè cercare di ricreare una nuvola di data point con una forma quanto più simile a quella originaria, t-SNE si concentra nell'evidenziare le relazioni di similarità locali tra i vari sample. Utilizzare un algoritmo di riduzione della dimensionalità diametralmente opposto a quello della PCA consente di ottenere informazioni differenti da quelle già raccolte; inoltre, qualora i risultati ottenuti fossero coerenti tra i due algoritmi, fornirebbero istruzioni chiare su come effettuare la fase di modellazione.

Il corretto utilizzo di t-SNE è piu complesso di quello della PCA, in quanto bisogna fare particolare attenzione al setting degli iperparametri. In particolare,

deve essere posta molta attenzione sul numero massimo di iterazioni e sul valore di perplexity.

Il numero massimo di iterazioni indica un upper bound alle iterazioni eseguibili per ottenere la convergenza. La convergenza di t-SNE viene ottenuta quando la matrice dei coefficienti di similarità dello spazio embedded (quindi, a minore dimensionalità) risulta essere uguale alla matrice dei coefficienti di similarità dello spazio originario, garantendo che, nella riduzione della dimensionalità, siano state mantenute le relazioni locali tra i punti.

Nel dataset di riferimento, il problema della scelta del numero massimo di iterazioni non si pone, in quanto, essendo il dataset relativamente piccolo, già solo dopo 1000 - 2000 iterazioni viene raggiunta la convergenza. Per questa ragione, il team di lavoro ho posto tale parametro a 3000 iterazioni. Per quanto riguarda, invece, il valore della perplexity, il discorso è sicuramente piu complesso. Tale iperparametro fornisce un'indicazione sulla densità di punti attesa per ciascun cluster; ovviamente, il modello finale è molto sensibile al valore di perplexity scelto. Sono stati eseguiti numerosi test, partendo da un valore di perplexity molto basso, pari a 2, fino ad arrivare a 20. I risultati sono stati i seguenti:

- con valore di perplexity basso (tra 2 e 5) sono ben visibili tra i 3 ed i 4 cluster (Figura 5.10);
- all'aumentare del valore di perplexity, il numero di cluster visibili diminuisce fino ad arrivare a 2 (Figura 5.11);
- con una perplexity superiore a 10, i due cluster non sono più distinguibili, unendosi in un insieme piuttosto uniforme di punti (Figura 5.12).

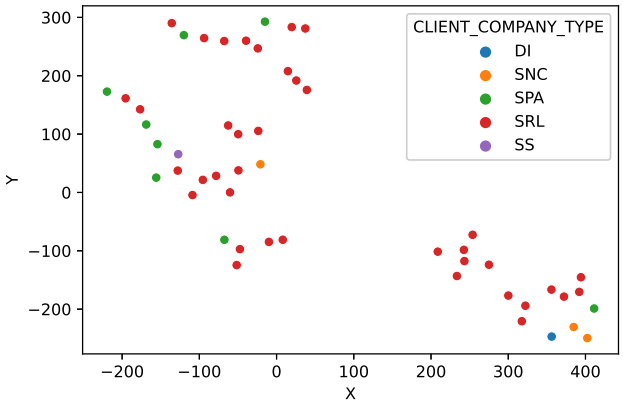

Figura 5.10. t-SNE con perplexity pari a 3

Esattamente come con la PCA, i risultati di t-SNE ci permettono di individuare facilmente almeno 2 cluster, ed il loro numero può salire fino a 3 o 4 se si considerano rilevanti le differenze locali tra punti adiacenti. Sfortunatamente, l'algoritmo di t-SNE non ci consente di fare considerazioni relative alla densità e alla distanza tra i

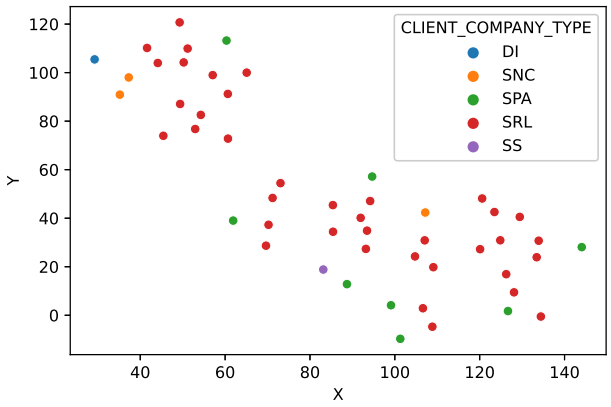

Figura 5.11. t-SNE con perplexity pari a 7

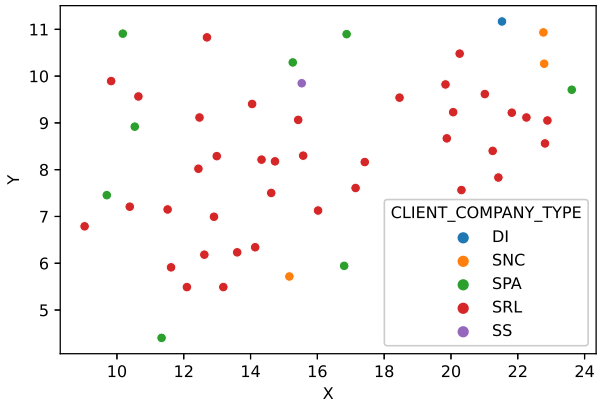

Figura 5.12. t-SNE con perplexity pari a 15

vari cluster, in quanto, durante la trasformazione dallo spazio originario allo spazio embedded, viene soltanto garantita la conservazione delle distanze locali tra i data point.

6

Attività di Clustering e Analisi dei Centroidi

In questo capitolo è riportata la fase di modellazione più importante ed impegnativa che il team di lavoro ha affrontato durante l'intero progetto. Vengono descritte le differenti tecniche di clustering che sono state adottate, specificando, gli approcci utilizzati per il setting degli iperparametri. Successivamente, i risultati dei differenti algoritmi vengono confrontati tra loro per decidere le label definitive da assegnare a ciascuna azienda. Infine, è presente un'analisi dei centroidi eseguita per assegnare un significato finanziario a ciascun cluster.

6.1 DBSCAN

DBSCAN individua dei cluster nelle zone di maggiore densità, classificando i data point come core point (centroidi dei cluster), border point (punti periferici di un cluster) e noise point (rumore). A differenza di altri algoritmi, DBSCAN non richiede in input il numero dei cluster previsti, ma necessita di un attento tuning degli iperparametri che definiscono l'intorno di ciascun data point.

6.1.1 Tuning degli iperparametri

Gli iperparametri di DBSCAN da dover scegliere per poter ottenere dei risultati corretti sono:

- *eps* - la epsilon dell'intorno, quindi la lunghezza del raggio della circonferenza, centrata in ciascun data point, che permette di individuare i punti vicini;
- *min_samples* - il numero minimo di punti vicini per poter considerare il data point centrale come un core point.

Fortunatamente, è possibile trovare il migliore valore di *eps* in maniera indipendente dal valore di *min_samples*, utilizzando la tecnica del Nearest Neighbor.

Quest'ultima consiste nel calcolare, per ogni data point, la distanza al data point immediatamente più vicino. Ordinando tale distanze e disegnandole su un grafico, è stato ottenuto il risultato in Figura 6.1.

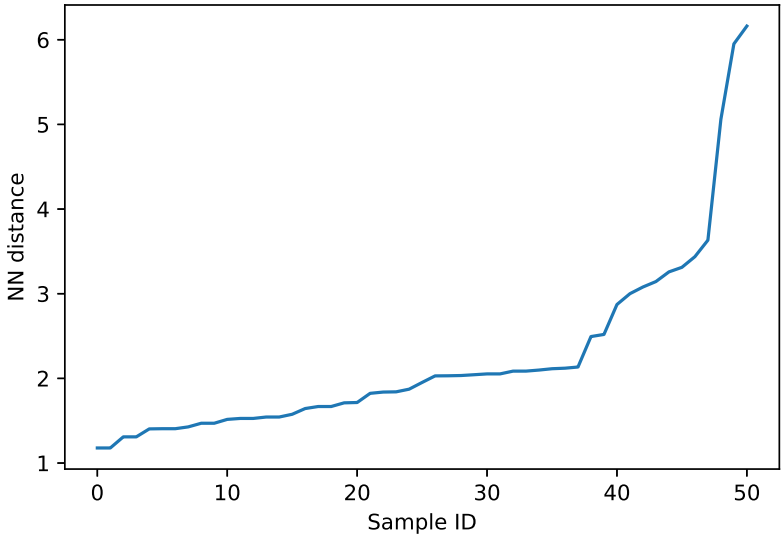

Figura 6.1. Nearest Neighbor distance di ogni sample

Dal grafico è possibile notare come quasi 40 sample abbiano una Nearest Neighbor distance minore di 2.15; vi è, poi, un forte aumento della distanza tra gli ultimi sample.

Tale risultato è coerente con quanto avevamo notato visivamente dal plot della PCA, in cui la maggior parte dei sample erano descritti dalle prime due componenti principali, mentre un altro piccolo gruppo era sparso lungo la terza.

Il valore di epsilon scelto è stato, quindi, di 2.15.

Una volta scelto *eps*, calcolare il miglior valore di *min_samples* è stato relativamente semplice, perchè è bastato realizzare 51 test, cambiando il valore dell'iperparametro da 1 a 51, per individuare quel test che permettesse di trovare il maggior numero di cluster minimizzando il numero di punti non classificati (quindi, quelli etichettati come noise point).

Il miglior valore di *min_samples* è risultato essere 6.

6.1.2 Risultati

L'attendo tuning degli iperparametri ci ha consentito di trovare 2 cluster, da 21 e 14 punti, e di avere soltanto 16 punti etichettati come noise point.

Il numero dei punti non classificati da DBSCAN non deve scoraggiare, in quanto essi sono proprio i punti con un valore di Nearest Neighbor distance molto più grande rispetto alla media. Ciò significa che, piuttosto che essere soltanto rumore, è possibile che essi siano a loro volta raggruppati in un terzo cluster, ma, essendo esso meno denso degli altri due (perchè le Nearest Neighbor distance tra i punti sono molto più grandi), DBSCAN non riesce a raggrupparli insieme.

Per poter indagare maggiormente su questa terza nuvola di punti, e per poter confermare l'affidabilità dei risultati di DBSCAN sugli altri due gruppi, è stato necessario utilizzare un ulteriore algoritmo di clustering, ovvero K-Means.

6.2 K-Means

K-Means è un algoritmo molto diverso da DBSCAN; esso, infatti, prova a separare i data point in K gruppi di ugual varianza, minimizzando una misura che indica la distanza intra cluster, chiamata inerzia.

I K centroidi vengono inizialmente posizionati randomicamente nello spazio delle feature, per poi essere spostati in modo da ottenere dei risultati migliori.

6.2.1 Tuning degli iperparametri

Dato il non determinismo dell'algoritmo, causato dalle posizioni iniziali dei centroidi scelte randomicamente, è consigliabile eseguire l'algoritmo di clustering ripetendo un numero elevato di test, in modo da ottenere in output solo il risultato che ha ottenuto il minore risultato di inerzia, e, quindi, dei cluster di output più coesi.

Con il dataset di riferimento, ogni singola istanza di output di K-Means è stata ottenuta eseguendo 1000 test, scegliendo tra esse quella con la migliore performance.

Per scegliere il miglior valore dell'iperparametro K (corrispondente al numero dei cluster attesi) sono stati calcolati due diversi indicatori, con valori di K compresi tra 2 e 10: il *Sum of Squared Error (SSE)* ed il *Silhouette coefficient*. I risultati sono visibili in Figura 6.2 e Figura 6.3.

Per l'*SSE*, il miglior valore di K è quello corrispondente al gomito della sua curva, mentre per il *Silhouette Coefficient* il migliore valore di K è quello che corrisponde al massimo della sua curva.

In base alle informazioni ottenute dalle precedenti analisi, il numero dei cluster è supposto essere di almeno pari a 2, possibilmente pari a 3, e difficilmente maggiore.

Tuttavia, su suggerimento sia dell'*SSE* che del *Silhouette Coefficient*, sembrerebbe portare ad un valore superiore, di circa pari a 5.

In realtà, i grafici sono abbastanza fuorvianti, in quanto il miglioramento ottenuto nei valori dei due indicatori, passando da 3 a 5 cluster, è veramente minimo. Inoltre, è risultato che i 2 nuovi cluster sono, in realtà, costituiti da singoli outlier. Queste motivazioni ci hanno portato a scegliere un valore di K pari a 3.

6.2.2 Risultati

L'attento tuning degli iperparametri ci ha consentito di individuare 3 cluster da 26, 17 e 8 punti ciascuno.

6.3 Confronto dei risultati

La similitudine nelle dimensioni dei cluster ottenuti dai due algoritmi è, sicuramente, un risultato incoraggiante.

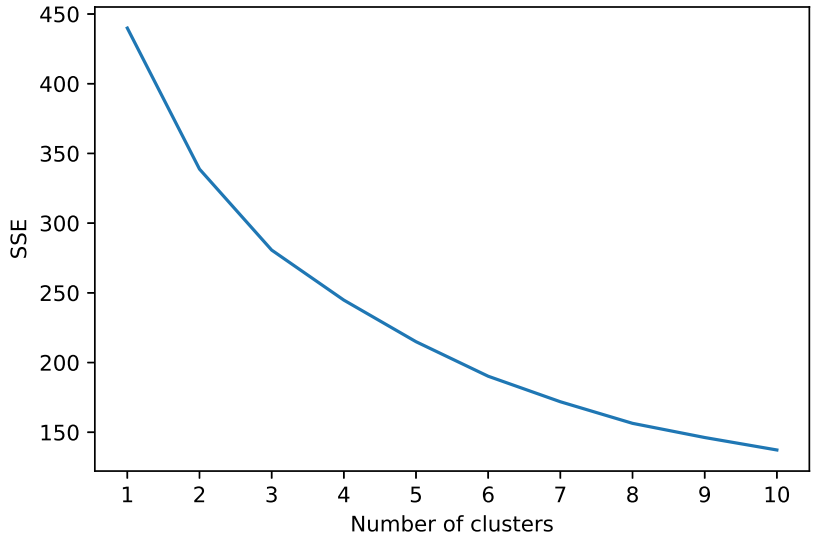

Figura 6.2. SSE al variare di K

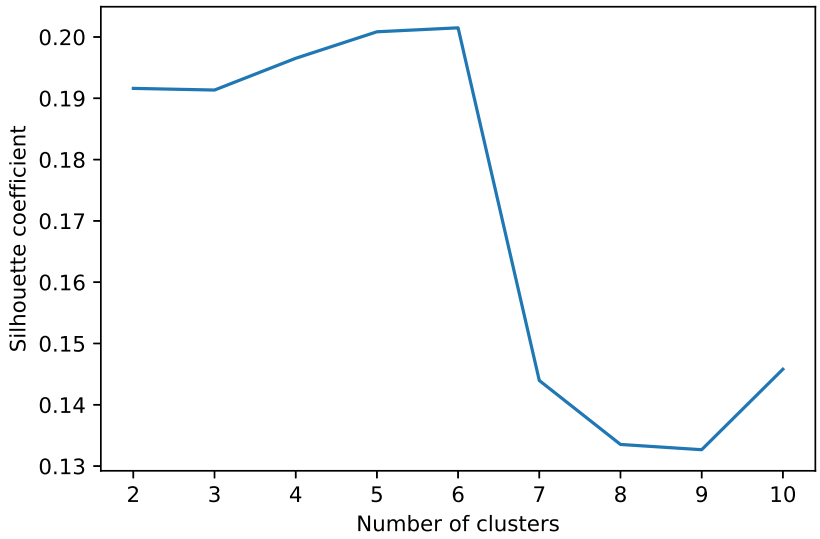

Figura 6.3. Silhouette coefficient al variare di K

Per poter avere la conferma della coerenza dei risultati ottenuti, si è deciso di confrontare le label dei cluster, in modo da poter essere certi che i cluster individuati fossero effettivamente gli stessi. Si è scelto di utilizzare il metodo grafico dei

diagrammi di Venn, per poter confrontare tra loro coppie di cluster corrispondenti e, quindi, con simili dimensioni, relativi ai due differenti algoritmi, evidenziando le possibili intersezioni. I risultati sono riportati nelle Figure 6.4, 6.5 e 6.6.

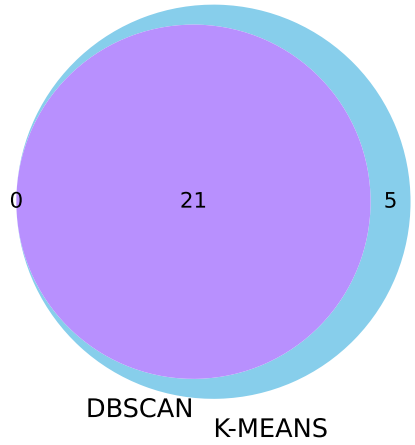

Figura 6.4. Confronto delle label dei cluster di maggiori dimensioni

Entrambi i cluster principali evidenziano una elevata similarità nei due algoritmi, in quanto i punti individuati da DBSCAN sono quasi sempre contenuti anche nel corrispondente cluster di K-Means.

Per quanto riguarda, invece, i 16 punti non classificati da DBSCAN, essi sono stati confrontati con il terzo cluster di K-Means.

La metà dei punti non classificati da DBSCAN sono contenuti all'interno del cluster di K-Means con minori dimensioni, dimostrando che l'intuizione sulla presenza di un cluster a minore densità era corretta.

6.3.1 Scelta delle label

L'attento tuning degli iperparametri ci ha consentito di ottenere dei risultati coerenti tra i due algoritmi. La validità dei risultati è stata mostrata graficamente utilizzando dei diagrammi di Venn, che riportano intersezioni di grandi dimensioni, o, addirittura, inclusioni, tra coppie di cluster corrispondenti.

Ciò ci ha permesso di assegnare una label a ciascun data point utilizzando il seguente criterio:

1. I punti clusterizzati nello stesso gruppo da entrambi gli algoritmi sono stati inseriti nello stesso cluster.

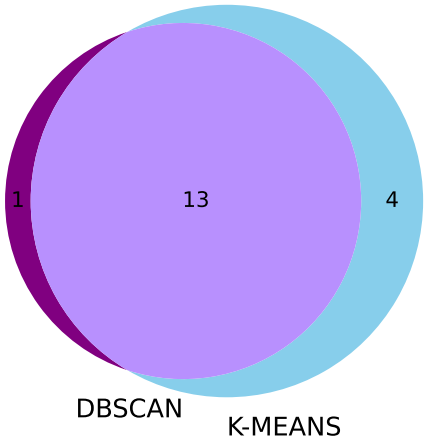

Figura 6.5. Confronto delle label dei cluster di medie dimensioni

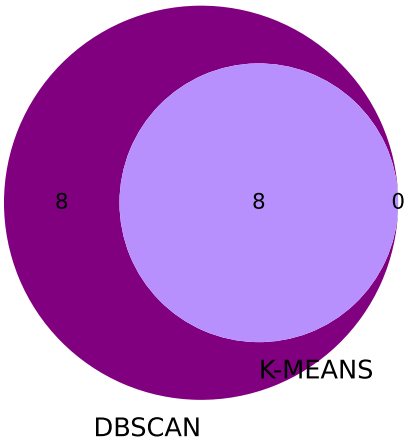

Figura 6.6. Confronto delle label dei cluster di minori dimensioni

2. I pochi punti classificati dai due algoritmi in gruppi differenti sono stati etichettati dando priorità alla decisione di DBSCAN, in quanto il criterio di densità è stato ritenuto più affidabile di quello di vicinanza ai centroidi.

3. I restanti 8 punti non classificati da DBSCAN, che non appartenevano al terzo cluster di K-Means, sono stati assegnati ai primi due cluster sulla base della decisione presa da K-Means.

6.4 Analisi dei centroidi

Suddivisi tutti i data point nei tre cluster, la fase successiva è stata quella di confrontare tra loro le medie e le mediane delle feature tra i vari cluster, al fine di trovare un significato finanziario che distinguesse i punti di cluster differenti. Tale analisi viene denominata analisi dei centroidi.

Per una più facile lettura, i valori delle feature più significative sono state normalizzate e mostrate nella heat map di Figura 6.7.

	cluster	0	1	2	std
total_negative_days	mean	0.119139	0.517225	0.669798	0.232134
total_positive_days	mean	0.047095	0.274112	0.465736	0.171113
unusual_positive_activity	mean	0.111111	0.200000	0.375000	0.109627
CLIENT_REVENUES	mean	0.474393	0.496553	0.716827	0.109436
num_transactions	median	0.005736	0.046800	0.214509	0.090308
expenses_concentration	mean	0.361939	0.311734	0.475024	0.068290
revenues_concentration	mean	0.578788	0.414556	0.509137	0.067304
unusual_negative_activity	mean	0.222222	0.320000	0.375000	0.063181
hist_negative_ratio	mean	0.323757	0.321074	0.243665	0.037139
hist_usage_ratio	mean	0.478215	0.476296	0.542162	0.030608
recent_revenues	mean	0.632455	0.638483	0.575866	0.028205
recent_expenses	median	0.267232	0.266322	0.225329	0.019543
trend_expenses	mean	0.631853	0.632564	0.671666	0.018603
median_balance	median	0.149486	0.151259	0.187694	0.017608
trend_revenues	mean	0.424747	0.416409	0.385499	0.016883
trend_balance	mean	0.368165	0.340249	0.374020	0.014735
total_ratio	median	0.098592	0.103424	0.078439	0.010820

Figura 6.7. Heat map dei centroidi

Analizzando tale heat map possiamo fare le seguenti osservazioni:

- il valore di *client_revenues* permette di fare una netta distinzione tra il cluster 2 ed i cluster 0 ed 1;
- i valori di *total_positive_days* e di *total_negative_days* aumentano all'aumentare dell'ID del cluster;
- il valore di *num_transactions* aumenta all'aumentare dell'ID del cluster;
- i valori di *unusual_positive_activity* e di *unusual_negative_activity* aumentano all'aumentare dell'ID del cluster;
- il valore di *revenues_concentration* è molto alto nel cluster 0;
- il valore di *expenses_concentration* è molto alto nel cluster 2;
- il valore di *hist_negative_ratio* aumenta al diminuire dell'ID del cluster.

Da tali osservazioni, è possibile definire il significato finanziario di ciascun cluster nel seguente modo:

- **Cluster 2** - Aziende con elevato fatturato, che effettuano un elevato numero di transazioni, sia in entrata che in uscita. Sono caratterizzate da pochi grandi fornitori, ma da un maggior numero di clienti. Esse sono, sicuramente, le aziende più attive ed in buona salute.
- **Cluster 0** - Aziende diametralmente opposte a quelle del Cluster 2. Esse sono caratterizzate da un basso fatturato e da un minore numero di transazioni, le quali risultano essere maggiormente in uscita che in ingresso. Tali aziende hanno pochi clienti. Esse sono le aziende meno attive dal punto di vista finanziario e, perciò, instabili.
- **Cluster 1** - Aziende intermedie, più vicine al Cluster 0 che al Cluster 2, ma relativamente attive e più stabili di quelle del Cluster 0.

7

Indicatori di rischio

In questo capitolo viene descritta la realizzazione di due differenti indicatori di rischio, uno data driven, ed uno rule based. Questi indicatori intermedi verranno, poi, uniti insieme in un modello bayesiano per ottenere l'indicatore di rischio finale.

7.1 Progettazione degli indicatori di rischio

L'informazione più importante ottenuta fino a questa fase del progetto è l'etichetta di ciascuna azienda associata ad un particolare cluster, ovvero il cluster delle aziende instabili (label con valore pari a 0), quello delle aziende stabili (label con valore pari a 2), ed, infine, quello delle aziende intermedie (label con valore pari a 1). Tale informazione da sola non soddisfa, però, i requisiti del progetto, che richiedono un indicatore numerico di rischio.
A tal proposito, il team di lavoro si è interrogato sulla modalità di calcolo di tale indicatore, anche a fronte delle informazioni ottenute dalla clusterizzazione. Si è optato per la realizzazione di due differenti indicatori di rischio, un indicatore data driven ed uno rule based. Essi sono indicatori intermedi, che verranno, poi, uniti insieme tramite un modello statistico bayesiano per ottenere l'indicatore finale.
Il motivo per il quale si è deciso di definire due indicatori è perchè, in questo modo, è possibile quantificare il rischio di credito ad un'impresa utilizzando due informazioni differenti, una legata ai dati, e quindi al cluster di appartenenza, l'altra alle regole di finanza quantitativa.

7.2 Indicatore data driven

L'indicatore data driven viene calcolato sulla base delle informazioni ottenute dal clustering. Si è deciso di associare il rischio di un'azienda alla sua probabilità di appartenere al cluster delle aziende instabili. Il problema, però, è che entrambi gli algoritmi utilizzati per la clusterizzazione forniscono in output delle hard label binarie, senza nessun riferimento alla probabilità o alla distanza di ciascun punto dai tre cluster. Per ovviare a questo limite, si è deciso di eseguire un bootstrap degli algormiti di clustering.

7.2.1 Bootstrap

In data analytics, il termine "bootstrap" viene spesso utilizzato per indicare tecniche o algoritmi differenti; in statistica esso viene definito formalmente come una tecnica di ricampionamento con reimmissione per approssimare la distribuzione campionaria di una statistica. Nel nostro caso, esso è particolarmente utile per stimare proprio la probabilità che ha ciascun campione di appartenere al cluster delle aziende instabili. Il bootstrap è stato implementato nel seguente modo:

1. Su ogni campione, è stata applicata 10 000 volte la fase di clustering precedentemente descritta nel Capitolo 6, salvando ogni volta l'etichetta associata a quell'azienda.
2. Il non determinismo del K-Means farà si che le etichette non siano sempre esattamente le stesse, definendo, quindi, una certa fluttuazione statistica sull'etichettatura di ciascun campione.
3. Al termine dell'esecuzione di tutti i test, si ottiene come risultato una lista di 10 000 label associate ad ogni azienda. Contando le occorrenze delle etichette di ciascun cluster, e dividendo per 10 000, si ottiene la probabilità che ha ogni azienda di essere associata ai tre cluster.
4. L'indicatore di rischio data driven è la probabilità di appartenere al cluster delle aziende instabili.

7.3 Indicatore rule based

L'indicatore rule based viene calcolato come somma pesata dei valori di un sottoinsieme di KPI ritenuti particolarmente utili a descrivere la stabilità finanziaria di un'azienda.

Il team di lavoro ha interrogato gli esperti finanziari di Virtual B per poter comprendere quali fossero, tra quelli già calcolati, i KPI da utilizzare ed i pesi da associare a ciascuno di essi.

I risultati forniti dai consulenti sono riportati in Figura 7.1.

KPI	Peso
1 - monthly_balance.12	0.3
expenses_concentration	0.1
revenues_concentration	0.4
total_ratio	0.1
1 - trend_balance	0.1

Figura 7.1. KPI e pesi utilizzati per calcolare l'indicatore rule based

Pesi elevati sono stati associati al bilancio dell'ultimo mese e all'indice di concentrazione dei clienti. Per quanto riguarda il bilancio, tanto maggiore è il suo valore e tanto minore è il rischio di credito all'azienda; per quanto riguarda l'indice di concentrazione, tanto maggiore è il suo valore, e tanto minore è il numero di clienti su cui fa affidamento l'azienda, evidenziando, quindi, la sua instabilità. L'indice di concentrazione dei fornitori, il coefficiente della regressione lineare dei bilanci mensili

ed il rapporto tra il numero di giorni con saldo negativo e quelli con saldo positivo, hanno un peso minore, ma sono sufficientemente rilevanti da essere considerati nel calcolo dell'indicatore di rischio rule based.

7.4 Confronto dei risultati

Si è ritenuto interessante confrontare tra loro i valori dell'indicatore data driven e di quello rule based di ogni azienda del dataset. I risultati sono mostrati nello scatter plot di Figura 7.2.

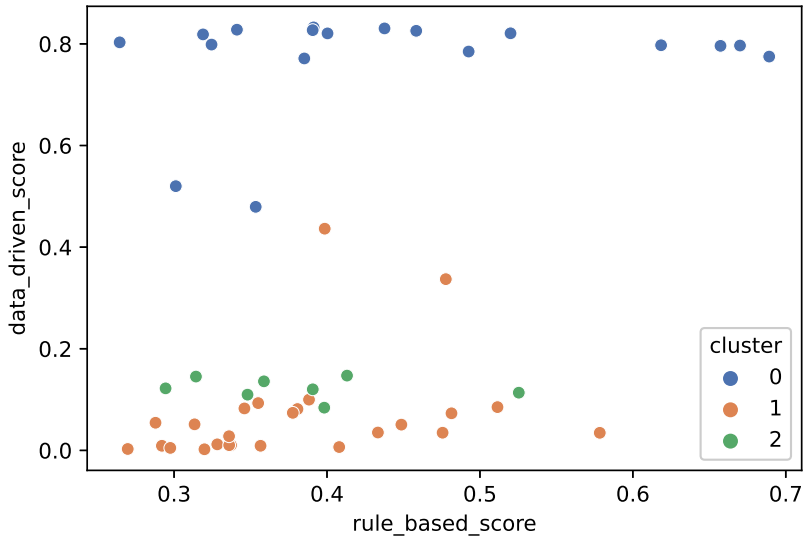

Figura 7.2. Scatter plot dei due indicatori di rischio

Valori uguali o simili dei due indicatori sono riportati lungo la diagonale principale del grafico, mentre valori differenti sono riportati in basso a destra e in alto a sinistra.

I data point sono colorati in base al cluster a cui appartengono.

Possiamo notare dallo scatter plot che molte aziende che hanno il valore di un indicatore molto basso hanno anche molto basso il valore dell'altro. Esistono, poi, 4 aziende con entrambi gli indicatori caratterizzati da valori molto elevati. Inoltre, è possibile individuare le seguenti anomalie:

- 3 aziende hanno il valore dell'indicatore data driven molto basso (<0.2) ma il valore di quello rule based alto (>0.5);
- 7 aziende hanno il valore dell'indicatore data driven molto alto (>0.7) ma il valore di quello rule based basso (<0.4).

La presenza di queste anomalie non dovrebbe scoraggiare, anzi, esse sono prova del fatto che sono necessari entrambi gli indicatori per poter valutare tutti i fattori che determinano la stabilità economica di un'impresa. L'elevata presenza di anomalie in alto a sinistra nel grafico, e l'assenza di punti in basso a destra, mostrano la tendenza dell'indicatore data driven ad assegnare valori di rischio alti, mentre quello rule based tende a sottostimare il rischio.

8

Definizione e utilizzo di un modello bayesiano

In questo capitolo viene introdotto il modello bayesiano, un modello statistico utilizzato per il calcolo dell'indicatore di rischio finale. Per fare ciò, vengono prima calcolate le distribuzioni degli indicatori intermedi di rischio.

8.1 Modello bayesiano

Calcolati i due indicatori intermedi di rischio, una semplice operazione potrebbe essere quella di effettuare una somma pesata dei due, in modo da ottenere uno score riassuntivo delle diverse informazioni. Tale soluzione, nonostante sia efficace, è stata scartata dal team di lavoro in favore dell'utilizzo di un modello bayesiano. Un modello bayesiano è un modello statistico che si basa sul teorema di Bayes (Figura 8.1).

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

Figura 8.1. Teorema di Bayes

Tale teorema afferma che la probabilità condizionata di A dato B è pari al prodotto tra la probabilità condizionata di B dato A e la probabilità di A, il tutto diviso per la probabilità di B.

Nella versione generale del teorema, A e B sono due eventi, ma spesso, nel contesto della data science, vengono sostituiti come ipotesi e dati, ottenendo la formula di Figura 8.2.

In questa versione, la probabilità dell'ipotesi condizionata ai dati viene chiamata posterior, la probabilità dei dati condizionata all'ipotesi viene detta likelihood, la probabilità dell'ipotesi è la prior, mentre la probabilità dei dati può essere chiamata marginal, evidence o total probability.

Tale intepretazione del teorema di Bayes, detta diacronica, fornisce un modo per aggiornare nel tempo la probabilità di un'ipotesi, sulla base della lettura di nuovi dati di input.

$$P(H|data) = \frac{P(data|H)P(H)}{P(data)}$$

Figura 8.2. Interpretazione diacronica del teorema di Bayes

Nel nostro progetto, il valore dell'indicatore rule based può essere visto come la prior, quello dell'indicatore data driven come la likelihood, e l'indicatore di rischio finale come posterior.

Per poter definire questo modello bayesiano è, però, necessario calcolare, per ogni singola azienda, le distribuzioni dell'indicatore data driven e di quello rule based.

8.2 Distribuzioni dell'indicatore data driven

Per ottenere la distribuzione dell'indicatore data driven, il team di lavoro non ha fatto altro che ripetere 10 000 volte l'operazione eseguita per calcolare lo score associato ad ogni azienda. In questo modo, si può ottenere una distribuzione di 10 000 valori per ogni azienda. Il problema, però, è che ogni valore della distribuzione richiede, a sua volta, 10 000 iterazioni per essere calcolato. Ciò determina un numero di iterazioni totali pari a 100 milioni che, in un normale laptop, non sono riproducibili. Anche abbassare il numero di data point a 1000 non porta evidenti miglioramenti; per tale ragione sono stati calcolati, alla fine, solo 100 punti per ogni singola distribuzione, impiegando circa due ore di tempo di esecuzione.

L'istogramma di una delle 51 distribuzioni calcolate è visibile in Figura 8.3.

8.3 Distribuzioni dell'indicatore rule based

A differenza dell'indicatore data driven, il calcolo per quello rule based non ci consente di estrarre nessuna fluttuazione statistica per costruire una distribuzione, essendo tale calcolo una semplice media pesata. Il team di lavoro ha, quindi, deciso di ricreare una varianza nei valori utilizzando il setting dei pesi associati alle feature. Nella versione base, ciascuna delle 5 feature utilizzate nella media pesata è associata ad un particolare peso, scelto dai consulenti finanziari sulla base di regole finanziarie; è, però, possibile introdurre un'incertezza nei pesi utilizzando una distribuzione, anzichè uno scalare. Tali distribuzioni verranno generate automaticamente tramite il modulo `NumPy`, passando come medie delle distribuzioni proprio i pesi precedentemente scelti, e come varianze il valore 0.1, ritenuto sufficientemente grande da ottenere una fluttuazione statistica. Essendo le distribuzioni dell'indicatore data driven limitate a soli 100 punti, si è dovuto imporre questo limite anche per quelle rule based. Le distribuzioni dei pesi sono state unite insieme per formare la matrice B, mentre i valori delle 5 feature rilevanti sono stati inseriti nella matrice A.

Calcolando il prodotto scalare tra le matrici A e B si ottiene la matrice C, che contiene le 51 distribuzioni dell'indicatore rule based (Figura 8.4).

L'istogramma di una delle 51 distribuzioni calcolate è visibile in Figura 8.5.

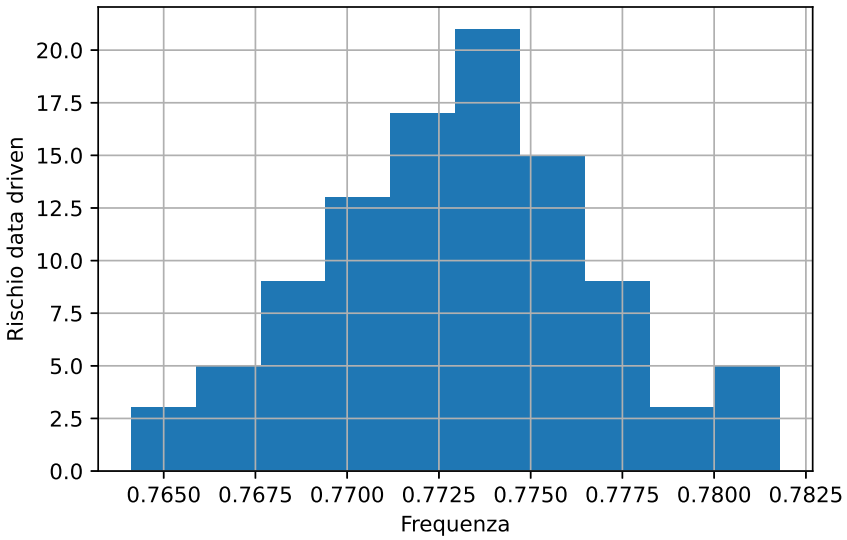

Figura 8.3. Istogramma di una delle distribuzioni data driven

$$\underset{51\times100}{C} = \underset{51\times5}{A} \bullet \underset{5\times100}{B}$$

Figura 8.4. Prodotto cartesiano per calcolare la matrice delle distribuzioni rule based

8.4 Famiglia di distribuzioni coniugate

Nell'ambito della teoria della probabilità bayesiana, se la distribuzione a posteriori è della stessa famiglia della distribuzione a priori, le due distribuzioni sono definite coniugate, e la distribuzione a priori è chiamata distribuzione a priori coniugata per la likelihood.

Ad esempio, la famiglia delle distribuzioni gaussiane è coniugata a se stessa rispetto ad una likelihood gaussiana; se, quindi, la likelihood fosse una distribuzione gaussiana, scegliere una distribuzione a priori gaussiana assicurerebbe che anche la distribuzione a posteriori sia gaussiana.

Lavorare con una famiglia di distribuzioni coniugate è conveniente dal punto di vista algebrico, in quanto fornisce un'espressione in forma chiusa per la distribuzione a posteriori; alternativamente, sarebbe necessario il calcolo, molto più complesso, di un integrale numerico.

Quindi, se le distribuzioni di prior e likelihood appartenessero alla stessa famiglia di coniugate, la teoria bayesiana fornirebbe delle semplici formule algebriche per calcolare i parametri della loro posterior.

Il team di lavoro ha analizzato con attenzione le distribuzioni data driven, ed ha concluso che potessero essere riconducibili a delle distribuzioni simil-gaussiane. Per quanto riguarda le distribuzioni rule based, esse sono state generate come di-

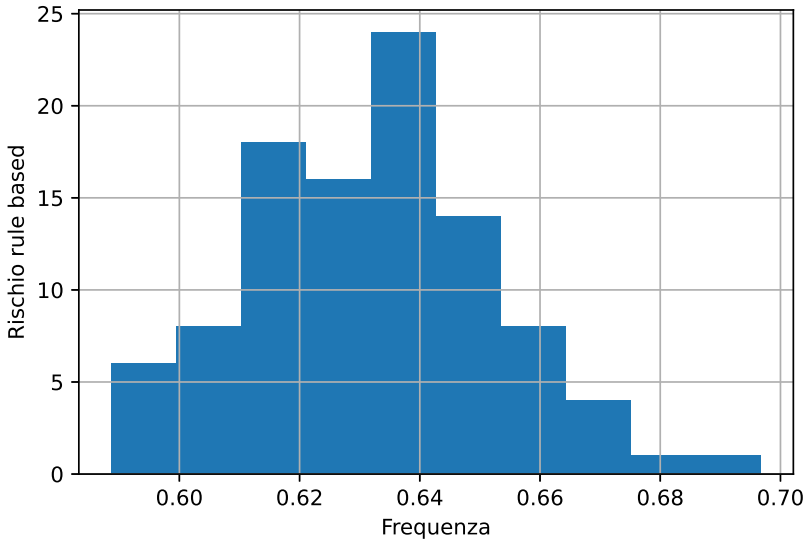

Figura 8.5. Istogramma di una delle distribuzioni rule based

stribuzioni triangolari, le quali sono considerabili in letteratura come una buona approssimazione di distribuzioni gaussiane.

8.4.1 Indicatore di rischio finale

Riassumendo, i passi eseguiti per calcolare l'indicatore di rischio finale di ogni azienda, aggregando le distribuzioni data driven e rule based, sono stati i seguenti:

1. Calcolo della distribuzioni data driven.
2. Calcolo della distribuzione rule based.
3. Utilizzo della teoria bayesiana per associare la distribuzione data driven ad una likelihood (nota), la distribuzione rule based ad una prior (nota) e la distribuzione dell'indicatore aggregato come posterior (da calcolare) delle due.
4. Calcolo diretto della media della posterior, supponendo la prior e likelihood come appartenenti alla stessa famiglia di coniugate (in particolare, sono state considerate come gaussiane).
5. La media della posterior rappresenta l'indicatore di rischio finale dell'azienda.

$$m'' = \frac{\sigma^2 m' + n\sigma'^2 m}{n\sigma'^2 + \sigma^2}$$

Figura 8.6. Formula per il calcolo della media della posterior con prior e likelihood guassiane

Nella Figura 8.6 viene riportata la formula per il calcolo della media della posterior, utilizzata come indicatore di rischio finale. I parametri m, σ^2 ed m', $\sigma^{2,}$ sono la media e la varianza, rispettivamente, di likelihood e prior. Molto interessante è il parametro n, calcolato come rapporto tra il numero dei punti della likelihood e la prior; esso permette di definire in modo differente il contributo delle due distribuzioni sulla base della loro numerosità campionaria. Una distribuzione definita a partire da un campione ridotto è meno attendibile di una costruita con un campione più numeroso; di conseguenza, nel calcolo finale, dovrebbe essere pesata di meno rispetto all'altra, per cercare di ridurre l'errore nella stima della posterior.

Il team di lavoro ha deciso di costruire sia la prior che la likelihood con la stessa numerosità campionaria (100 misure); di conseguenza, ha potuto utilizzare il parametro n per assegnare un peso differente alle due distribuzioni sulla base dell'affidabilità delle informazioni da loro veicolate. Si è deciso di effettuare diversi test, cambiando i valori dei pesi da assegnare alle due distribuzioni, per mostrare poi gli istogrammi dei risultati, visibili nelle Figure 8.7, 8.8 e 8.9.

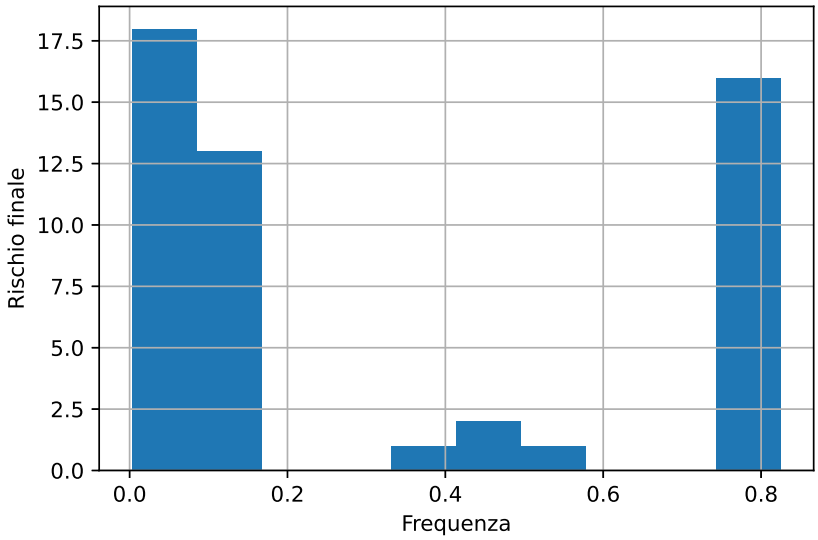

Figura 8.7. Istogramma del rischio finale con peso data driven pari a 0.8 e peso rule based pari a 0.2

Dagli istogrammi è possibile notare come la distribuzione data driven tenda a binarizzare i valori del rischio finale (rischio alto o rischio basso), mentre, quella rule based, permetta di individuare anche delle situazioni di aziende con rischio medio, le quali potrebbero essere analizzate con maggiore attenzione da un operatore umano. Per tale ragione, il team di lavoro ha deciso di dare maggior peso alla distribuzione rule based.

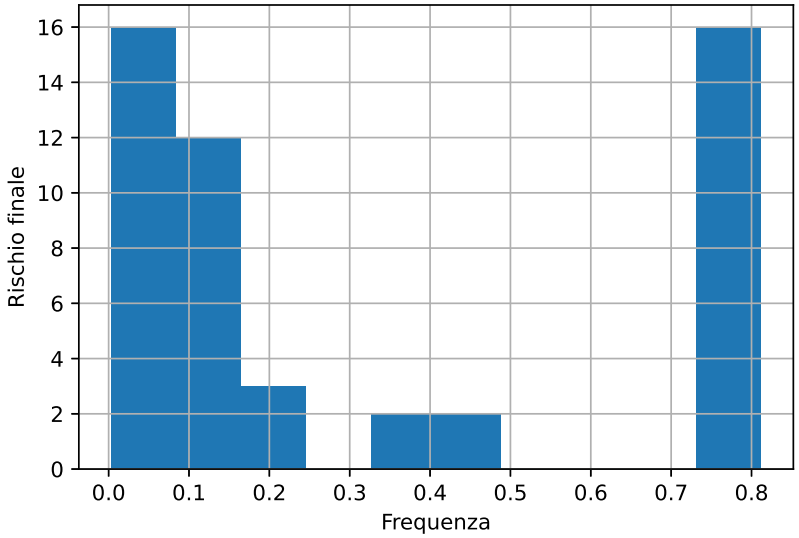

Figura 8.8. Istogramma del rischio finale con peso data driven pari a 0.5 e peso rule based pari a 0.5

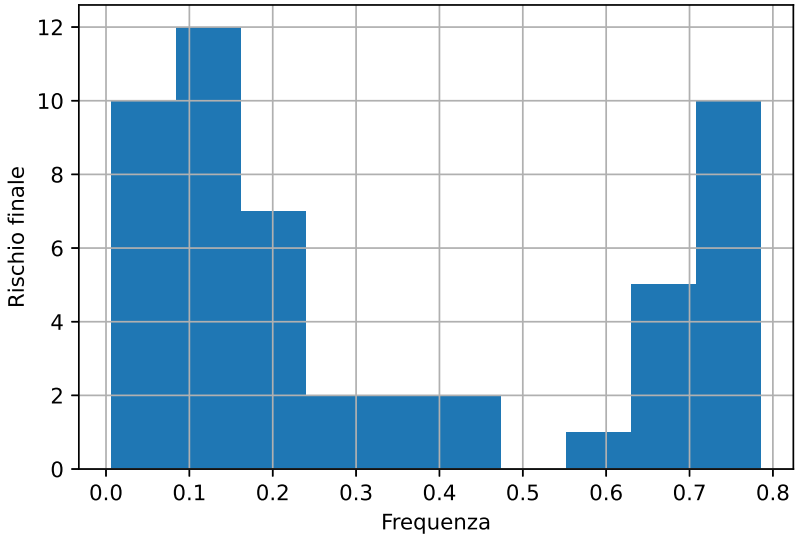

Figura 8.9. Istogramma del rischio finale con peso data driven pari a 0.2 e peso rule based pari a 0.8

9

Network Analysis

In questo ultimo capitolo di analisi viene descritta una modellazione delle relazioni tramite grafo. In particolare, le relazioni che hanno i clienti della banca con i loro fornitori ed i loro clienti vengono quantificate con degli indicatori numerici. Questi ultimi vengono aggregati insieme per ottenere un indicatore finale, il quale consente di identificare i clienti con un network di relazioni particolarmente grande ed interconnesso.

9.1 Premessa

Il dataset iniziale ci permette di trovare, per ogni cliente della banca, i fornitori ed i clienti con i quali ha eseguito delle transazioni sul proprio conto corrente.

La distinzione tra clienti e fornitori viene effettuata guardando la direzione della transazione; se è un versamento, allora il soggetto è un cliente, altrimenti è un fornitore.

Oltre ad utilizzare queste informazioni per calcolare degli indicatori di concentrazione tramite i KPI, il team di lavoro ha pensato di sfruttarle per eseguire una Network Analysis, quindi un'analisi su grafo delle relazioni esistenti tra ciascun cliente della banca, i suoi fornitori ed i suoi clienti.

Una piccola nota riguarda la nomenclatura utilizzata in questo capitolo. Fino ad ora, si è utilizzato il termine "cliente" per indicare, semplicemente, i clienti della banca; tuttavia, nei prossime sezioni, verranno analizzati anche i clienti dei clienti della banca. Per evitare ambiguità, verrà utilizzata la seguente nomenclatura:

- *NDG (Numero Direzione Generale)* - i clienti della banca;
- *clienti* - i clienti degli NDG;
- *fornitori* - i fornitori degli NDG.

9.2 Struttura del grafo

La prima cosa da fare quando si effettua una Network Analysis è decidere che tipo di struttura utilizzare per il grafo. Nel dataset di riferimento, è possibile fare le seguenti osservazioni:

- i nodi sono di tre tipi diversi (NDG, clienti e fornitori);
- i nodi fornitori e clienti possono essere collegati solo con i nodi NDG;
- i nodi NDG possono essere collegati solo con i nodi clienti e i nodi fornitori.

Sulla base di tali osservazioni, risulta naturale la scelta di un grafo tripartito per modellare le relazioni.

Tuttavia, il team di lavoro ha deciso di non utilizzare direttamente il grafo tripartito, preferendo invece dividerlo in due grafi bipartiti, in quanto i risultati delle analisi risultano, in questo modo, essere più facilmente interpretabili.

Sono stati, quindi, realizzati i seguenti grafi bipartiti:

- grafo bipartito NDG - clienti;
- grafo bipartito NDG - fornitori.

9.2.1 Grafo bipartito NDG-clienti

Il grafo bipartito NDG-clienti è costituito di 51 nodi NDG e 2513 nodi clienti, collegati insieme da 2581 archi. Essendo il grafo bipartito, la sua densità è bassa, pari a circa 0.02.

Nonostante la bassa densità del grafo, il numero relativamente elevato di nodi e di connessioni non consente una sua chiara visualizzazione grafica.

9.2.2 Grafo bipartito NDG-fornitori

Il grafo bipartito NDG-fornitori è costituito di 51 nodi NDG e 5963 nodi fornitori, collegati insieme da 6130 archi. Essendo il grafo bipartito, la sua densità è bassa, pari a circa 0.02.

Nonostante la bassa densità del grafo, il numero relativamente elevato di nodi e di connessioni non consente una sua chiara visualizzazione grafica.

9.3 Indicatori calcolati

Le analisi su grafo si traducono, generalmente, nel calcolo di indicatori che forniscono misure quantitative delle relazioni tra i nodi.

Prima di mostrare e commentare i risultati delle analisi svolte, viene riportata, di seguito, una breve descrizione degli indicatori calcolati.

- *Degree* - è, semplicemente, il numero di archi incidenti su un nodo; maggiore è il numero di archi e maggiore è l'importanza che quel nodo ha nella rete.
- *Clustering coefficient* - indica quanto coesi siano i vicini di un nodo. I valori estremi sono 0, se il nodo forma con i propri vicini una stella, ed 1, se forma una clique, cioè un insieme di nodi totalmente connessi.
- *Degree centrality* - è la misura di degree normalizzata, calcolata facendo il rapporto tra il numero di archi incidenti sul nodo ed il numero di nodi totali, equivalente al numero massimo di archi possibili su quel nodo. Tuttavia, nel caso di grafo bipartito, essa viene calcolata in modo leggermente diversa, in quanto il numero massimo archi incidenti su un nodo non è pari al numero di nodi totali nel grafo, bensì al numero di nodi dell'altro insieme.

- *Closeness centrality* - fornisce una misura di vicinanza di un nodo a tutti gli altri nodi della rete. Intuitivamente, si potrebbe pensare che un nodo con una elevata degree centrality abbia automaticamente una closeness centrality alta, ed in effetti in molti grafi i due concetti si sovrappongono; tuttavia, nei grafi bipartiti, le due forme di centralità rimangono sempre concetti differenti.
- *Betweenness centrality* - permette di trovare i cosiddetti nodi "bridge", cioè nodi che fanno da collegamento tra due o più cluster di nodi ben distinti.
- *Eigenvector centrality* - individua quelle che, nel gergo comune, vengono chiamate "eminenze grigie", cioè individui molto influenti ma poco visibili. La visibilità di un nodo è data dal suo numero di connessioni, quindi la eigenvector centrality associa un valore più alto a quei nodi connessi a nodi con una degree molto elevata.

9.4 Analisi delle relazioni

I due grafi sono stati analizzati separatamente, calcolando i vari indicatori per ogni nodo. Sono stati, poi, filtrati solo i risultati dei nodi NDG, escludendo i valori degli indicatori per i nodi clienti ed i nodi fornitori, in quanto ritenuti non interessanti.

9.4.1 Grafo bipartito NDG-clienti

I valori di degree centrality, closeness centrality, betweenness centrality, eigenvector centrality e clustering coefficient di ogni nodo NDG sono stati normalizzati e mostrati sulla heat map di Figura 9.1, per poter avere una rappresentazione grafica intuitiva dei risultati.

Osservando la heat map, è possibile fare le seguenti osservazioni:

- L'NDG 888884940956 risulta ricoprire un ruolo molto importante nella rete con i clienti, avendo il valore più elevato di degree, betweenness ed eigenvector centrality di tutti gli altri nodi. Esso è sia l'NDG con il maggior numero di clienti che un importante nodo bridge nei rapporti tra differenti coppie di NDG e clienti. Un basso valore di closeness centrality indica il fatto che esso non è un NDG tipico nella rete, ma, anzi, svolge un ruolo "vip". Nonostante esso sia distante dal centro della rete, il suo valore di eigenvector centrality molto elevato indica che i clienti a cui è collegato sono, a loro volta, collegati ad un grande numero di altri NDG, sottolineando ulteriormente la sua importanza sociale nella rete.
- L'NDG 888886437522 è secondo per numero di clienti e betwenness centrality, seguito da 888888213271, 888884950265, e 888888013088.
- La closeness centrality ha valori molto alti per circa la metà degli NDG, i quali, a loro volta, sono caratterizzati da bassi valori di degree e betweenness centrality, ciò indica che vi è un gran numero di aziende medie con reti sociali con i clienti piuttosto uniformi tra loro.
- Molto interessante è l'NDG 888887359660, il quale, pur avendo bassi valori in tutti gli altri indicatori, ha il valore più alto di clustering coefficient. Ciò significa che i pochi clienti che ha sono molto coesi tra loro (formano, cioè, delle clique). Un discorso simile vale per gli NDG 888889588232, 888889987936 e 888888833742.

	degree	closeness	betweenness	eigenvector	clustering
888884448922	0.010787	0.362140	0.000430	0.000000	0.166468
888884460100	0.000000	0.000000	0.000000	0.000000	0.000000
888884475589	0.001079	1.000000	0.000000	0.000000	0.000000
888884653066	0.125135	1.000000	0.003842	0.000000	0.000000
888884670505	0.015102	1.000000	0.000052	0.000000	0.000000
888884679597	0.025890	0.171401	0.028374	0.000002	0.343137
888884696771	0.003236	1.000000	0.000002	0.000000	0.000000
888884940956	1.000000	0.388879	1.000000	1.000000	0.028425
888884950265	0.122977	0.264652	0.132635	0.004944	0.110588
888884974525	0.019417	1.000000	0.000088	0.000000	0.000000
888884997258	0.014024	0.219908	0.014842	0.001094	0.015424
888884998757	0.004315	1.000000	0.000003	0.000000	0.000000
888886022875	0.043150	0.258119	0.048405	0.001144	0.263639
888886221465	0.006472	0.269216	0.010399	0.002203	0.526755
888886437522	0.430421	0.242704	0.455209	0.000007	0.032895
888886536940	0.007551	1.000000	0.000012	0.000000	0.000000
888887118792	0.000000	0.000000	0.000000	0.000000	0.000000
888887118900	0.018339	0.258490	0.019771	0.001122	0.313017
888887359660	0.007551	0.175165	0.010145	0.000007	1.000000
888887822955	0.084142	0.822430	0.002173	0.000000	0.166468
888887901496	0.010787	1.000000	0.000026	0.000000	0.000000
888888013088	0.101402	0.259612	0.123738	0.003618	0.150897
888888107049	0.020496	0.221947	0.020367	0.001108	0.369732
888888213271	0.173679	0.289555	0.276073	0.001314	0.047921
888888335648	0.051780	0.265500	0.058357	0.005712	0.208117
888888398390	0.030205	0.294786	0.181333	0.001143	0.309082
888888622829	0.017260	1.000000	0.000069	0.000000	0.000000
888888757271	0.070119	0.262267	0.078538	0.001185	0.164137
888888764324	0.002157	1.000000	0.000001	0.000000	0.000000
888888780763	0.014024	0.187549	0.019799	0.000010	0.394157
888888793688	0.004315	1.000000	0.000003	0.000000	0.000000
888888833742	0.024811	0.222820	0.032292	0.003321	0.601500
888888973983	0.032362	0.268612	0.040290	0.003363	0.162436
888889055977	0.016181	0.211446	0.047312	0.000006	0.348805
888889308891	0.001079	1.000000	0.000000	0.000000	0.000000
888889328639	0.002157	1.000000	0.000001	0.000000	0.000000
888889363350	0.001079	1.000000	0.000000	0.000000	0.000000
888889371861	0.019417	0.261821	0.022780	0.001130	0.387045
888889517653	0.032362	0.271592	0.043854	0.003375	0.234145
888889541634	0.010787	0.258057	0.011139	0.001113	0.378304
888889547233	0.009709	0.165654	0.011601	0.000000	0.500551
888889551796	0.003236	1.000000	0.000002	0.000000	0.000000
888889588232	0.006472	0.227527	0.004986	0.002181	0.627957
888889688565	0.001079	1.000000	0.000000	0.000000	0.000000
888889745453	0.005394	1.000000	0.000006	0.000000	0.000000
888889786414	0.030205	0.233597	0.056671	0.002244	0.299085
888889805542	0.012945	1.000000	0.000038	0.000000	0.000000
888889840151	0.085221	1.000000	0.001774	0.000000	0.000000
888889842836	0.031284	0.278907	0.126315	0.001125	0.274930
888889899841	0.015102	0.193271	0.014046	0.000013	0.417336
888889987936	0.007551	0.255002	0.008053	0.001104	0.616915

Figura 9.1. Heat map degli indicatori del grafo NDG-clienti

9.4.2 Grafo bipartito NDG-fornitori

I valori di degree, closeness, betweenness ed eigenvector centrality, nonchè quelli del clustering coefficient di ogni nodo NDG, sono stati normalizzati e visualizzati sulla heat map di Figura 9.2, per poter avere una rappresentazione grafica intuitiva dei risultati.

	degree	closeness	betweenness	eigenvector	clustering
888884448922	0.066298	0.075952	0.047477	0.001237	0.274054
888884460100	0.001105	1.000000	0.000000	0.000000	0.000000
888884475589	0.012155	1.000000	0.000009	0.000000	0.000000
888884653066	0.001105	1.000000	0.000000	0.000000	0.000000
888884670505	0.103867	0.081089	0.077831	0.001293	0.218670
888884679597	0.139227	0.129808	0.100174	0.002693	0.141755
888884696771	0.017680	1.000000	0.000019	0.000000	0.000000
888884940956	1.000000	0.236831	1.000000	1.000000	0.071520
888884950265	0.272928	0.115629	0.201364	0.010788	0.118648
888884974525	0.040884	1.000000	0.000099	0.000000	0.000000
888884997258	0.276243	0.151448	0.212008	0.001708	0.092999
888884998757	0.040884	0.026088	0.030060	0.000006	0.054587
888886022875	0.054144	0.095269	0.045449	0.003593	0.243885
888886221465	0.083978	0.143108	0.068955	0.003740	0.179532
888886437522	0.092818	0.134631	0.069884	0.002606	0.211250
888886536940	0.050829	0.127405	0.037343	0.001251	0.184287
888887118792	0.069613	0.143831	0.058863	0.002507	0.223873
888887118900	0.130387	0.129566	0.094638	0.001371	0.124098
888887359660	0.012155	0.060417	0.008878	0.000037	0.441460
888887822955	0.191160	0.083111	0.137336	0.004138	0.071700
888887901496	0.006630	0.000000	0.004888	0.000001	0.764214
888888013088	0.017680	0.103077	0.021203	0.003457	0.400992
888888107049	0.062983	0.049857	0.046228	0.000009	0.270358
888888213271	0.045304	0.130932	0.032955	0.003624	0.288982
888888335648	0.449724	0.168067	0.343479	0.010273	0.074680
888888398390	0.176796	0.195932	0.208289	0.009622	0.162107
888888622829	0.034254	0.091717	0.030654	0.001188	0.371660
888888757271	0.056354	0.133018	0.043218	0.001285	0.210728
888888764324	0.085083	0.134382	0.068796	0.001302	0.169950
888888780763	0.058564	0.142387	0.050588	0.003658	0.247003
888888793688	0.012155	0.062732	0.008957	0.001127	0.900220
888888833742	0.035359	0.191372	0.201821	0.005945	0.413931
888888973983	0.035359	0.118703	0.042561	0.004686	0.281357
888889055977	0.078453	0.092024	0.056709	0.002479	0.221834
888889308891	0.000000	0.074698	0.000000	0.001145	0.526396
888889328639	0.001105	0.062542	0.000815	0.001115	1.000000
888889363350	0.002210	0.018581	0.001630	0.000001	0.363002
888889371861	0.355801	0.145754	0.257366	0.003682	0.104148
888889517653	0.150276	0.083621	0.107600	0.000088	0.143432
888889541634	0.545856	0.150314	0.418749	0.005176	0.069578
888889547233	0.019890	0.016854	0.014648	0.000007	0.413203
888889551796	0.019890	0.102391	0.026887	0.000091	0.311719
888889588232	0.188950	1.000000	0.002080	0.000000	0.000000
888889688565	0.001105	0.063589	0.000005	0.002226	0.306239
888889745453	0.454144	0.082294	0.321538	0.006106	0.061355
888889786414	0.411050	0.102112	0.297856	0.001987	0.087782
888889805542	0.005525	0.062182	0.003292	0.001119	0.125112
888889840151	0.265193	0.158589	0.216414	0.007756	0.134899
888889842836	0.290608	0.111454	0.213444	0.003262	0.093405
888889899841	0.140331	0.163727	0.148778	0.007960	0.210149
888889987936	0.053039	0.072941	0.037564	0.003514	0.224858

Figura 9.2. Heat map degli indicatori del grafo NDG-fornitori

Osservando la heat map, possiamo fare le seguenti osservazioni:

- In questo grafo, il principale nodo “vip” (cioè, con un valore elevato di degree, betweenness ed eigenvector centrality) risulta essere sempre l’NDG 888884940956, seguito da 888889541634, 888889745453, 888888335648 e 888889786414;
- Il numero di nodi NDG con un elevato valore di closeness centrality è minore

rispetto all'altro grafo, probabilmente perchè si ha un numero di nodi fornitori doppio rispetto al numero dei nodi clienti, risultando più difficile, per i nodi NDG, essere centrali nel grafo bipartito;
- I nodi NDG con un clustering coefficient elevato (quindi i cui fornitori sono maggiormente coesi tra loro) sono 888889328639, 888888793688, 888887901496 e 888889308891.

9.5 Valutazione dei risultati

A prescindere dal possibile interesse e dalla possibile curiosità che possono suscitare i risultati di una Network Analysis, si deve sempre valutare la loro utilità nello specifico business case.

Innanzitutto, le heat map riportano i valori normalizzati dei risultati degli indicatori dei soli nodi NDG, escludendo, per una più chiara e semplice trattazione, i valori dei migliaia di nodi clienti e fornitori. Da una più attenta analisi dei valori non normalizzati si capisce che, in realtà, i valori di clustering coefficient e di closeness centrality degli NDG sono troppo bassi per essere ritenuti rilevanti.

Diverso è, invece, il discorso per i valori di degree e di betweenness centrality, che hanno consentito di individuare nelle due reti i cosiddetti nodi "vip". I loro valori sono sufficientemente elevati, anche considerando i valori di tutti gli altri nodi clienti e fornitori, da poterli considerare effettivamente rilevanti.

I nodi "vip" precedentemente identificati sono riportati nella Figura 9.3.

Grafo bipartito NDG-clienti	Grafo bipartito NDG-fornitori
888884940956	888884940956
888886437522	888889541634
88888213271	888889745453
888884950265	888888335648
888888013088	888889786414

Figura 9.3. Nodi "vip" individuati

Nonostante il primo nodo "vip" sia comune ad entrambi i grafi, tutti gli altri nodi risultano "vip" solo in uno dei due grafi, a dimostrazione del fatto che la divisione del grafo tripartito iniziale in due grafi bipartiti differenti ha effettivamente facilitato l'interpretazione dei risultati d'analisi. Tali informazioni potrebbero essere utilizzate dai manager della banca per scegliere dei potenziali promotori finanziari, oppure per altre finalità legate al networking.

9.6 Indicatori di networking finali

Il team di lavoro ha ritenuto che i differenti indicatori calcolati tramite la Network Analysis non fossero sufficientemente chiari ed intuitivi per il cliente finale del

progetto, si è, quindi, deciso di realizzare due indicatori aggregati (uno per ogni grafo) che riassumessero tutte le informazioni raccolte. Tali indicatori, chiamati *clients_network_score* e *suppliers_network_score*, sono stati calcolati come somma pesata degli indicatori normalizzati dei grafi corrispondenti. Si è dato un peso elevato alla degree e alla betweenness centrality (0.3), un peso medio al clustering coefficient (0.2), ed un peso basso alla closeness centrality e all'eigenvector centrality (0.1), per le motivazioni descritte nel paragrafo precedente.

Con la scelta di tali pesi, i nodi individuati come "vip" dall'analisi della heat map risultano avere i valori aggregati più alti di tutti, permettendo una loro veloce e facile identificazione.

In Figura 9.4 è riportato lo scatter plot dei due indicatori.

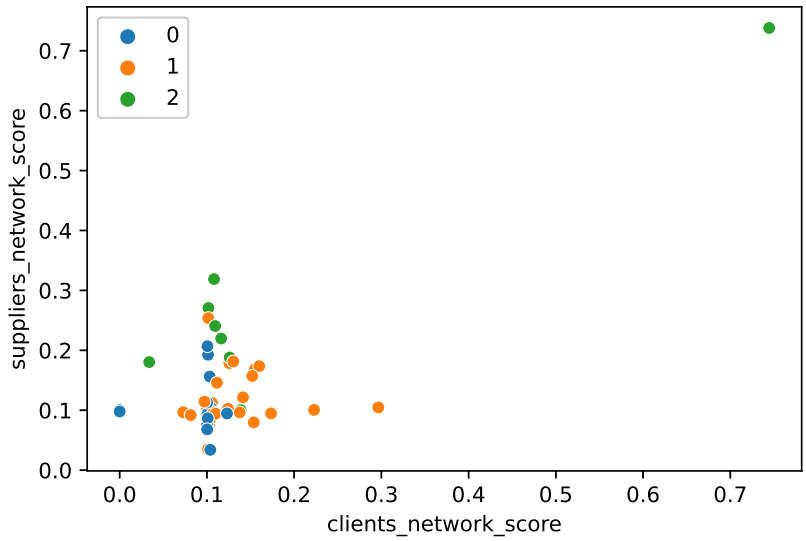

Figura 9.4. Scatter plot degli indicatori finali di Network Analysis

Da tale scatter plot, si nota subito il nodo "vip" con relazioni di clienti e fornitori molto più rilevanti della media. Inoltre, è interessante notare come le aziende con un *clients_network_score* più alto siano aziende del cluster 1, mentre quelle con un *suppliers_network_score* più alto siano principalmente aziende del cluster 2.

10

Considerazioni in merito al lavoro svolto

In questo capitolo vengono presentate tutte le considerazioni personali in merito al progetto realizzato, all'azienda che ha offerto il tirocinio e al mercato del lavoro per i data scientist in Italia.

10.1 Considerazioni sul progetto

Il tirocinio si è rivelato una grande opportunità per poter comprendere come un progetto di data analytics venga effettivamente organizzato ed eseguito in un'azienda. La fase di visualizzazione dei risultati in una dashboard per il cliente finale è stata eseguita da una ditta esterna, consentendo al team di lavoro di concentrarsi unicamente sulle fasi di analisi e di modellazione.
In particolare, abbiamo compreso quanto importante sia l'analisi delle distribuzioni e la comprensione delle tecniche statistiche di base, che spesso vengono tralasciate in ambiente universitario, in favore di corsi sul machine learning. Quest'ultimo, nonostante sia stato molto utile per l'estrazione di informazioni "data driven", è sempre stato applicato con molta attenzione e solo dopo un attento tuning degli iperparametri.
Il costante scambio di opinioni con i consulenti finanziari dell'azienda ci ha permesso di imparare come presentare e discutere i risultati con persone prive di un background ingegneristico.
Inoltre, il progetto ha portato anche alla realizzazione di un semplice framework di KPI di finanza quantitativa, il quale è stato documentato e reso disponibile per altri futuri progetti.
Infine, durante l'ultima settimana di lavoro, è stato chiesto al team la realizzazione di un semplice prototipo di app web che riproducesse le analisi principali del progetto. Il prototipo è stato realizzato utilizzando il moderno framework di Python `Streamlit`. Ciò ha dato la possibilità al team di lavoro di apprendere l'utilizzo di un facile strumento di prototipazione di app data-centered.

10.2 Considerazioni sull'azienda ospitante

L'anima della Virtual B, in parte finanziaria e in parte analitica, ci ha consentito di vivere appieno la realtà FinTech italiana. Il team, nonostante sia di ridotte dimensioni, si è rivelato essere interamente costituito da professionisti esperti ed appassionati del proprio lavoro. Inoltre, grande valore viene data all'iniziativa e alla creatività dei dipendenti, fornendo ai giovani neolaureati un luogo perfetto per proporre ed implementare nuove soluzioni.

10.3 Considerazioni sul lavoro del data scientist in Italia

La ricerca delle offerte di lavoro, i colloqui, ed infine l'esperienza di tirocinio, ci ha consentito di avere un'idea del lavoro del data scientist in Italia. Innanzitutto, è bene tenere presente che il data scientist è solo una delle diverse professioni nell'ambito della data analytics. In teoria, è possibile distiguere almeno tre differenti professioni:

- *Data engineer* - professionista con elevati skill di sviluppo software e di progettazione di sistemi distribuiti. Ha l'incarico di occuparsi delle pipeline che generano, trasferiscono e memorizzano grandi quantità di dati, spesso real time, in uno o più database.
- *Data scientist* - riceve i dati dal data engineer e li analizza per estrarne informazioni utili. Egli fa uso principalmente di tecniche statistiche e algoritmi di machine learning. L'output che produce sono in genere KPI, modelli statistici o report, che possono essere consegnati al data analyst, oppure presentati direttamente al cliente finale.
- *Data analyst* - ha come scopo principale quello di interfacciarsi con il cliente, realizzando delle dashboard con le informazioni fornite dal data scientist.

Il data engineer è richiesto solo da grandi aziende con esigenze particolari, che lavorano con grandi quantità di dati distribuiti. I professionisti con questa specializzazione sono relativamente pochi, ed hanno uno stipendio mediamente più alto degli altri.

Seppur le figure del data scientist e del data analyst siano abbastanza sovrapponibili tra loro, generalmente i compiti richiesti nella maggior parte delle offerte di lavoro per data scientist riguardano esclusivamente la realizzazione di dashboard e, inoltre, sono pubblicate da aziende di consulenza informatica, le quali non richiedono nemmeno una elevata preparazione tecnica ai loro candidati.

Le vere richieste per data scientist provengono, principalmente, da aziende FinTech, oppure da grandi multinazionali che desiderano investire nella data analytics per creare valore dai propri dati.

Indubbiamente, il mercato del lavoro italiano per i professionisti dei dati è indietro rispetto a tanti altri hub europei; situazione testimoniata anche dal numero di offerte di lavoro per data scientist pubblicate su LinkedIn (Figura 10.1).

Paese	Offerte di lavoro
UK	3432
Germania	3059
Benelux	2528
Francia	1099
Spagna	914
Italia	638

Figura 10.1. Numero di offerte di lavoro per data scientist pubblicate su LinkedIn - maggio 2021

11

Conclusioni e uno sguardo al futuro

In questa tesi sono state descritte tutte le fasi relative ad una campagna di data analytics a supporto delle attività di un gruppo bancario. Partendo da dati di transazioni sui conti correnti dei clienti della banca, aggregando i valori con KPI di finanza quantitativa è stato ottenuto un nuovo dataset, che ha consentito di fornire informazioni dettagliate sullo stato finanziario di ciascun cliente. Dopo aver analizzato le distribuzioni delle feature più interessanti, è stata applicata una riduzione della dimensionalità per poter esplorare la struttura dei data point nello spazio delle feature.
I risultati ci hanno spinto ad applicare degli algoritmi di clustering per raggruppare i clienti in tre differenti gruppi; quello delle aziende a rischio, quello delle aziende stabili e quello delle aziende intermedie. Sono stati, poi, definiti due differenti indicatori di rischio, uno "data driven" e uno "rule based".
L'indicatore data driven è stato definito con un bootstrap di K-Means, calcolando la probabilità che ha ogni azienda di appartenere al cluster delle aziende a rischio. Invece, per definire l'indicatore "rule based", è stata calcolata una somma pesata dei KPI ritenuti più significativi dal punto di vista finanziario. I due indicatori intermedi sono stati, infine, aggregati insieme in un modello statistico bayesiano, mappando quello "data driven" come likelihood, quello "rule based" come prior e quello finale come "posterior" del modello. Il risultato è stato un indicatore numerico che ingloba al proprio interno differenti tipi di informazioni, sia quelle derivanti dai dati che quelle ricavate dalle conoscenze del dominio applicativo.
Si è deciso, inoltre, di sfruttare le informazioni sui beneficiari e gli ordinanti delle transazioni per effettuare una Network Analysis. Essa ci ha consentito di calcolare, per ogni azienda, due indicatori di networking, uno per le relazioni con i fornitori e l'altro per quelle con i clienti. I valori di tali indicatori individuano le aziende con importanti reti sociali, il cui stato di salute finanziaria si propagherebbe velocemente a molte altre aziende della rete (siano esse fornitori o clienti).
I KPI calcolati, gli indicatori di rischio e gli indicatori di networking, sono stati forniti ai tecnici informatici della banca, i quali hanno realizzato una dashboard che riassumesse graficamente i risultati da mostrare ai manager.

Le analisi effettuate sono state fortemente limitate dal relativo piccolo numero di aziende del campione fornito, che comprende solo 51 unità. L'eventuale estensione futura del dataset potrebbe permettere il training di un classificatore, il quale

predirebbe il cluster di appartenenza di una nuova azienda a partire dai valori dei suoi KPI. Inoltre, bisogna tenere presente che le transazioni riportate sono relative al solo anno 2018; se venissero aggiunte anche le transazioni di altri anni, sarebbe possibile effettuare un'analisi temporale che fornirebbe informazioni molto interessanti sullo stato finanziario delle aziende relativo al periodo temporale. Infine, la banca, per motivi di privacy, ha totalmente oscurato la posizione geografica delle aziende; se si potessero ottenere tali dati, si potrebbe eseguire una geoanalitica delle aziende lungo tutto il territorio nazionale.

Ringraziamenti

Frequentare il corso di laurea magistrale mi ha permesso di vivere dei momenti meravigliosi con tanti colleghi e professori. Di seguito, ringrazio le persone che, più di tutte, mi sono state accanto in questi ultimi anni:

- I miei genitori, che mi hanno sempre spinto a non accontentarmi e di puntare in alto nella vita.
- Le mie splendide sorelle, che mi hanno sopportato con grande pazienza.
- Il mio fratellino Thor, che mi ha insegnato il vero significato della parola amore.
- I miei best bro Gigi e Balda, che mi motivano a diventare un uomo più forte ogni giorno.
- La mia fantastica collega Sara, senza la quale le giornate sarebbero state molto più noiose e grigie.
- Capt. Mattia, che ha guidato la sua ciurma con grande saggezza in tutti i progetti universitari.
- Morris, per essere un bravissimo storyteller ed un grande amico.
- Alexandra, che con la sua dolcezza riesce sempre a farmi stare bene.
- Ruxxar, Tormentanna, Dube e tutti gli NPC che mi hanno aiutato a sopravvivere nelle meravigliose avventure nel mondo di Siling.

Ed infine, per ultimo ma non per importanza, un ringraziamento particolare va al professore Domenico Ursino, la cui grande umanità ed elevata professionalità lo rendono un esempio di eccellenza per l'intera facoltà.

Riferimenti bibliografici

1. Clustering Coefficient. https://www.youtube.com/watch?v=K2WF4pT5pFY, 2014.
2. How to Use t-SNE Effectively. https://distill.pub/2016/misread-tsne/, 2016.
3. StatQuest: t-SNE, Clearly Explained. https://www.youtube.com/watch?v=NEaUSP4YerM, 2017.
4. Estimating Probabilities with Bayesian Modeling in Python. https://towardsdatascience.com/estimating-probabilities-with-bayesian-modeling-in-python-7144be007815, 2018.
5. StatQuest: K-means clustering. https://www.youtube.com/watch?v=4b5d3muPQmA&list=LL&index=8&t=36s, 2018.
6. StatQuest: Principal Component Analysis (PCA), Step-by-Step. https://www.youtube.com/watch?v=FgakZw6K1QQ, 2018.
7. Visualizing Data with Pairs Plots in Python. https://towardsdatascience.com/visualizing-data-with-pair-plots-in-python-f228cf529166, 2018.
8. Bayes theorem. https://www.youtube.com/watch?v=HZGCoVF3YvM&list=LL&index=7&t=68s, 2019.
9. DBSCAN Python Example: The Optimal Value For Epsilon (EPS). https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc, 2019.
10. In Depth: Principal Component Analysis . https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html, 2019.
11. Introduction to Factor Analysis in Python. https://www.datacamp.com/community/tutorials/introduction-factor-analysis, 2019.
12. t-SNE Python Example. https://towardsdatascience.com/t-sne-python-example-1ded9953f26, 2019.
13. Feature Scaling for Machine Learning. https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/, 2020.
14. K-Means Clustering in Python: A Practical Guide. https://realpython.com/k-means-clustering-python/, 2020.
15. Monte Carlo Simulation. https://www.youtube.com/watch?v=7ESK5SaP-bc, 2020.
16. Allen B. Downey. *Think Stats*. Green Tea Press, 2014.
17. Allen B. Downey. *Think Complexity*. Green Tea Press, 2016.
18. Allen B. Downey. *Think Bayes*. Green Tea Press, 2020.
19. Michael Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.
20. Robert Tibshirani Trevor Hastie and Jerome Friedman. *The Elements of Statistical Learning*. Springer, 2009.