UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA INDUSTRIALE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

Tesi Di Laurea Magistrale In Ingegneria Informatica

STUDIO DELLE APPLICAZIONI DI INTELLIGENZA ARTIFICIALE NELL'AMBITO SALES AND MARKETING E ANALISI CON STRUMENTI DI MACHINE LEARNING

Relatore: Prof. Moreno Muffatto

Laureando: GIOVANNI ZAMPIERI

ANNO ACCADEMICO 2018-2019

A Giuseppina

b

Abstract

Obiettivo

Per investitori, venture capitalist e business angel la scelta delle startup ed imprese in cui investire è critica. Ad oggi, si registra un alto tasso di insuccesso per gli investimenti. Questo fatto può essere correlato anche ai processi decisionali applicati, spesso basati su analisi qualitative e sull'intuizione frutto di una prolungata esperienza nel settore. Ad oggi, solo un ristretto numero di società di investimento dichiara di avvalersi di strumenti basati sul machine learning per la valutazione dei potenziali investimenti. Gli algoritmi utilizzati non sono tuttavia resi pubblici.
Il presente lavoro di tesi vuole analizzare le prestazioni di diversi strumenti di questo tipo, atti a supportare il processo decisionale di investitori early stage focalizzati su compagnie ed imprese *technology based.*

Metodologia

La base di dati a nostra disposizione è fornita dalla piattaforma Crunchbase. Il dataset utilizzato è il risultato di un'estrazione realizzata per focalizzare l'oggetto di studio su imprese operanti nell'area sales and marketing e facenti uso di strumenti di intelligenza artificiale e machine learning. Le imprese considerate costituiscono quindi un campione omogeneo per il settore.
I dati ottenuti dall'estrazione sono stati analizzati seguendo il workflow solitamente applicato in data science. Nella fase di preprocessing si è provveduto al cleaning dei dati, all'analisi delle feature esistenti, all'ingegnerizzazione di nuove feature e alla definizione delle variabili target. Volendo utilizzare classificatori binari, si è scelto di attribuire alla variabile target un valore positivo nel caso in cui l'impresa abbia effettuato una exit (in forma di acquisizione o Initial Public Offering, IPO), zero in caso di chiusura della stessa. Una volta prodotti un insieme di training ed un insieme di test, sono stati allenati e valutati comparativamente tre algoritmi noti in letteratura: Support Vector Machine (SVM), Neural Network (NN) e Random Forest (RF). L'implementazione dei modelli è stata effettuata utilizzando Python come linguaggio di programmazione.

Risultati

I modelli da noi testati hanno avuto successo nel processo di apprendimento: riescono infatti ad attribuire la label corretta con accuratezza tra il 70 ed il 75%. I risultati ottenuti possono quindi costituire un riferimento per ulteriori sviluppi futuri, aventi come oggetto ad esempio l'analisi di altri settori o l'applicazione di modelli alternativi, quali ad esempio il deep learning.

d

Indice

Indice ..... e
1 Introduzione 1
1.1 L'intelligenza artificiale ed il machine learning 1
1.1.1 Overview 1
1.1.2 Stato dell'arte 1
1.1 Il futuro dell'AI 4
1.1.1 Limiti e sfide 4
1.1.2 Sviluppi futuri 6
2 Applicazioni di AI nel settore sales and marketing 9
2.1 Nascita ed evoluzione dei servizi 9
2.1.1 Trend attuali e startup innovative 13
3 Dataset 17
3.1 Crunchbase 17
3.1.1 Storia 17
3.1.2 Crunchbase per la ricerca 18
3.1.3 Reperimento delle informazioni. 18
3.2 Descrizione dei campioni 19
4 Preprocessing dei dati 27
4.1 Selezione delle feature 30
4.1.1 Ottenere le coordinate geografiche tramite i servizi Google 31
4.1.2 Selezione della category group list con rete neurale 31
4.2 Cleaning dei dati 33
4.3 Feature Engineering 35
4.3.1 Burn rate 35
4.3.2 Variazione di burn rate 35
4.3.3 Investimenti ripetuti (fiducia degli investitori) 36
4.4 Statistica descrittiva 37
5 Metodologia ed ipotesi di ricerca 39
5.1 Domande di ricerca 39
5.2 Significatività della metodologia 40

5.3 Algoritmi utilizzati 41
5.3.1 K-means clustering 41
5.3.2 Support Vector Machine (SVM) 44
5.3.3 Random Forest 45
5.3.4 Neural Network 47
5.4 Ipotesi e risultati attesi 49
5.5 Metriche di valutazione 49
6 Risultati sperimentali 51
6.1 Selezione della grouplist 51
6.2 Clustering dei dati 55
6.3 Support Vector Machine (SVM) 57
6.4 Neural Network 61
6.5 Adaboost Random Forest 63
7 Analisi dei risultati 65
7.1.1 Clustering 65
7.1.2 SVM 65
7.1.3 Rete neurale 66
7.1.4 Random Forest 67
8 Conclusioni e sviluppi futuri 69
9 Indice delle figure 71
10 Indice delle tabelle 72
11 Riferimenti Bibliografici 73

1 Introduzione

1.1 L’intelligenza artificiale ed il machine learning

1.1.1 Overview

In questa tesi tratteremo di intelligenza artificiale e nello specifico di una sua branca: il *machine learning*, l’apprendimento automatico. Cosa significa “apprendere” per una macchina? Esattamente come per l’attività di apprendimento “naturale” tipica degli esseri viventi, per una macchina apprendere significa convertire esperienza in capacità decisionali. Significa, dato un certo input, poter agire autonomamente, sperabilmente in maniera corretta e utile, senza costante supporto e supervisione di un utente umano. Significa, ad esempio, saper riconoscere correttamente lettere e punteggiatura, a partire da un’immagine, e convertirli in un testo in formato digitale. Significa saper associare correttamente al suono delle parole, la relativa forma scritta. Non significa, invece, che le macchine siano in grado di leggere o ascoltare come un essere umano. I calcolatori sono in grado solo di eseguire operazioni matematiche, la cui logica e sequenza è nota quasi solo ai programmatori di ogni specifico sottoprogramma. Possiamo riferirci a questa logica con il termine comunemente usato di diagramma di flusso. In un diagramma di flusso, a partire da un input si arriva deterministicamente, attraverso una serie di scelte condizionate, ad un output. Un’intelligenza artificiale, avendo a disposizione uno o più diagrammi, non fa che processare l’input per produrre l’output.

Intuiamo quindi che apprendere automaticamente potrebbe significare, per le macchine, costruire in autonomia il proprio diagramma di flusso, utilizzando una forma di esperienza. Questo risulta molto utile quando le logiche che corrispondono alla relazione input - output sfuggono all’intuizione umana. Allora ciò di cui abbiamo bisogno è esperienza e metodo, ovvero un insieme di coppie input-output ed un *algoritmo di learning* che costruisca iterativamente e aggiorni un *modello* il cui obiettivo sarà classificare correttamente dei nuovi input. Di questo si occupa il machine learning: di fornire le strutture matematiche necessarie a questo compito. Negli anni sono stati proposti molteplici algoritmi, ne sono stati evidenziati i limiti e le potenzialità. Anche se gli studi teorici sono iniziati a metà del 900, solo nell’ultimo ventennio, con la crescente informatizzazione degli archivi dati e lo sviluppo della tecnologia dei computer, sono divenuti disponibili insiemi di input-output sufficientemente grandi e macchine sufficientemente potenti per sperimentare le potenzialità di questa teoria.

1.1.2 Stato dell’arte

L’intelligenza artificiale trova ogni giorno nuovi campi di applicazione nella ricerca, nell’industria e infine nella vita di tutti i giorni.

Nell'industria pesante, coadiuvare i macchinari con l'AI presenta molteplici vantaggi. Rende possibile l'automazione di nuovi compiti, permettendo l'operatività continua. Dove questo non è ancora possibile, migliora le interazioni uomo-macchina, riduce il costo operativo ed aumenta la sicurezza nell'ambiente di lavoro. Sono già largamente diffusi i sistemi di autodiagnostica, progettati per utilizzare il machine learning per prevedere ed intervenire tempestivamente su guasti e malfunzionamenti.

Nel settore dei trasporti, i veicoli a guida autonoma sono oggetto di importanti investimenti nella ricerca, mentre i primi modelli a guida assistita sono già disponibili sul mercato. Automatizzare completamente il processo di guida avrebbe importanti ricadute positive sull'economia e sulla qualità della vita a livello globale: ci si aspetta non solo una drastica riduzione nel numero degli incidenti stradali ma anche una diminuzione dell'inquinamento. Infatti, le auto a guida autonoma possono utilizzare le informazioni sul traffico per scegliere sempre i percorsi più convenienti e possono ottimizzare il consumo di carburante nelle fasi di partenza e accelerazione. Questo comporta una riduzione dell'emissione di agenti inquinanti quali poveri sottili e gas serra.

Il percorso verso la guida automatica è stato simbolicamente diviso in 5 step (BMW, 2019). È considerando come "livello 0" la guida completamente manuale, senza alcuna assistenza. Al livello 5 troviamo la guida completamente automatica in cui l'intelligenza artificiale del veicolo ha pieno controllo, mentre le persone al suo interno sono solo passeggeri.

Coadiuvata dall'intelligenza artificiale, la robotica sta facendo significativi passi avanti, in particolare, lo sviluppo di robot dalle sembianze umanoidi. Le sfide da affrontare in questo caso sono molte, tra cui il mantenimento dell'equilibrio, l'interazione con gli oggetti, con esseri umani ed eventualmente con altri robot. L'ente spaziale russo sta sperimentando l'impiego di robot umanoidi, i modelli FEDOR (Russian Foundation for Advanced Research Projects in the Defense Industry, 2017). Questi replicano i movimenti di un operatore remoto e sono in grado di eseguire spontaneamente alcuni compiti. Lo Skybot F-850, ha portato a termine una missione dimostrativa sulla ISS ad agosto 2019, anche se dai feedback degli astronauti emerge come la strada da percorrere sia ancora molta (MARINI, 2019).

Skybot F-850 ha avuto difficoltà ad interagire con il personale di bordo e nel muoversi all'interno della stazione è risultato goffo ed impacciato: Era ostacolato tra l'altro dalle sue stesse gambe, le quali, in un ambiente privo di gravità, si sono rivelate completamente inutili. Il robot è riuscito, dopo molti tentativi, ad eseguire autonomamente alcuni semplici compiti, come collegare dei cavi o adoperare un trapano. Queste operazioni potrebbero essere necessarie per eseguire delicati interventi di manutenzione e riparazione della stazione spaziale. Allo stato attuale dello sviluppo, i robot non sono pronti per la conquista dello spazio, ma nel lungo periodo il loro ruolo sarà fondamentale. Ad esempio, per il ruolo chiave nella costruzione di avamposti e basi lunari permanenti, la cui realizzazione, prevista intorno al 2050, dovrebbe iniziare nei prossimi anni (Baldacci, 2019).

In questo elaborato ci concentreremo sul lato software dell'AI, ovvero su quelle applicazioni che non richiedono la progettazione e realizzazione di hardware dedicato ma i cui servizi, basati per la maggior parte su architettura client-server, sono fruibili tramite dispositivi di uso comune, quali personal computer, smartphone e tablet. Proprio per la loro vasta accessibilità, questi servizi si stanno diffondendo rapidamente ed efficacemente.

Se parliamo di solo software, si entra nella categoria degli assistenti virtuali. Qui la comprensione e l'elaborazione del linguaggio naturale e dei dati in forma di testo, immagini, audio e video, nonché il riconoscimento delle emozioni dell'interlocutore o la capacità di apprendere e fornire raccomandazioni personalizzate, costituiscono il punto focale dell'offerta di servizi. Gli sviluppi vedono impegnati i colossi del settore: IBM con Watson (IBM, s.d.), Microsoft con Cognitive Toolkit (Microsoft, 2017) e Google con DeepMind (Google inc, s.d.), ma anche centinaia di startup che si specializzano per settori di mercato e applicazione.

A partire da queste piattaforme, vengono sviluppati servizi ad applicazioni. I più diffusi e noti al pubblico sono quelli detti *chatbot* o *chatter robot*, ovvero gli assistenti virtuali in grado di dialogare con noi. Vengono impiegati soprattutto sugli smartphone per recuperare informazioni utili alla vita quotidiana, per impostare la sveglia o un appuntamento in agenda, per inviare mail o sms, trovare luoghi o navigare o lanciare app. Tra i più noti, Siri di Apple (Apple inc, s.d.), Google Now (Google inc, s.d.), Cortana di Microsoft (Micorsoft inc, s.d.). Ma si stanno diffondendo anche i dispositivi per la domotica, di cui citiamo la celebre Alexa di Amazon (Amazon inc, s.d.).

Sempre più spesso quando si telefona presso un call center, in generale nell'ambito dell'assistenza clienti, il primo a risponderci è un assistente virtuale. Questi cercherà di trovare una soluzione al problema dell'utente senza l'intervento di un centralinista. Questi programmi sono definiti agenti cognitivi, capaci di adattarsi a nuove situazioni. Dotati della conoscenza di svariate lingue ed in grado di gestire contemporaneamente migliaia di conversazioni, possono rispondere alle domande dei clienti anche cercando su internet, se non conoscono preventivamente la risposta. Non solo, si documentano ed apprendono anche in base alle reazioni degli interlocutori. Capiscono, infatti, di che umore sono dal tono della voce e si comportano di conseguenza.

Svariate altre applicazioni sono possibili. Il sito *chatbots.org* registra più di 1200 assistenti virtuali basati sull'Intelligenza Artificiale, di questi 35 sono riconducibili ad aziende e startup italiane (Chatbots.org, s.d.).

1.1 Il futuro dell'AI

1.1.1 Limiti e sfide

L'apprendimento automatico è diverso dall'apprendimento fisico del cervello umano. È estremamente più potente ed efficace, in quanto i computer, sempre più sviluppati, possono eseguire sempre più operazioni in frazioni di secondo. Inoltre, i dati salvati non rischiano mai di essere dimenticati e possono essere ricopiati perfettamente un numero illimitato di volte a costo minimo. Così, il machine learning fornisce modelli che l'intuizione umana non avrebbe mai potuto individuare o, eventualmente, solo in un tempo inaccettabilmente lungo.

Tuttavia, l'apprendimento automatico è allo stesso tempo molto più limitato rispetto ad un essere umano, perché un computer è solo in grado di rilevare dei modelli. Inoltre, un modello, se scendiamo nel dettaglio, altro non è che una suddivisione di uno spazio vettoriale in sottoinsiemi. Il senso e la logica di questa divisione possono essere interpretati e validati solo dall'intelligenza umana. In ogni caso, è questa capacità di costruire autonomamente modelli, e contemporaneamente di individuare proprietà strutturali dei dati, a distinguere l'intelligenza artificiale dalle precedenti forme di computazione.

L'intelligenza artificiale può contare su di un supporto teorico molto raffinato ed elegante. Per analizzare ed esporre i limiti di questa tecnologia, partiamo dalla citazione di un'importante risultato della teoria: il *no free lunch theorem* (SHALEV-SHWARTZ & BEN-DAVID, 2014, p. 61 - 64). Questo teorema afferma e dimostra che non esiste un algoritmo di apprendimento universale, in grado di produrre un modello efficace per qualsiasi insieme di training. Nemmeno la rete neurale, il più potente e versatile degli algoritmi di machine learning.

È sempre necessaria una conoscenza a priori, un *bias* induttivo, un punto di partenza da fornire alla macchina. Supponendo di avere accesso ad un database, i progettisti dovranno innanzitutto definire il problema di learning. Ovvero, dovranno stabilire cosa vogliono, o cosa è utile, che la macchina impari.

Di conseguenza, attualmente è necessario affidarsi a soluzioni individuali per eseguire qualsiasi attività di marketing basata sull'IA. Dall'ottimizzazione e personalizzazione dei contenuti multimediali ad uno strumento gestione e ottimizzazione della spesa aziendale, ci sono molti strumenti di marketing intelligente tra cui scegliere. In mancanza di un *jack of all trades,* un *factotum*, occorre quindi testare e sperimentare con diverse soluzioni. Questo sforzo può non solo diventare dispendioso in termini di tempo e risorse, ma nel peggiore dei casi può concludersi con un nulla di fatto.

Altro problema è il costante bisogno di supervisione. La figura del marketer è ancora indispensabile per pianificare, progettare e gestire la campagna di marketing. In effetti, è da queste persone che proviene il punto di partenza, il *bias* induttivo, che viene fornito al sistema AI per apprendere. Questa forma di apprendimento supervisionato non imita il

modo in cui un essere umano impara naturalmente e gli esperti ritengono che questo sia uno dei maggiori ostacoli quando si tratta di implementare un'AI per l'interazione con gli umani.

Un celebre esempio che dimostra perché l'AI ha bisogno di supervisione è il chatbot di Microsoft Artificial Intelligence *"Tay"* (Tay Tweets, s.d.). Tay è stata progettata e modellata per imitare il comportamento di un'adolescente sui social, imparando proprio da altri adolescenti. Si era pensato che Tay sarebbe diventata sempre più intelligente e indistinguibile da un'adolescente, imparando dalle conversazioni da lei intrattenute con veri utenti umani. Tay era anche in grado di navigare il web ed accedere ai siti internet, così da potersi informarsi sui vari trend e topic ed arricchire ulteriormente le sue conoscenze. Era infine in grado di postare il risultato delle sue elaborazioni sul suo profilo Twitter.

Non sappiamo se gli ingegneri di Microsoft avessero preventivato la possibile presenza di utenti malintenzionati. Di certo, nessuno immaginava che l'AI sarebbe stata inondata di messaggi antisemitici, filonazisti e omofobi (La Stampa, 2106). Alcuni utenti non solo hanno suggerito a Tay frasi da ripetere, l'hanno anche spinta a cercare su Internet nuovo materiale per i suoi post. Alcuni dei discorsi di odio più coerenti postati da Tay, erano stati semplicemente copiati e adattati dal vasto archivio di contenuti antisemitici presente nel *dark web*. Meno 24 ore dopo il lancio, Microsoft fu costretta ad annullare il programma, nell'imbarazzo generale.

Questo tipo di incidenti ci aiuta a comprendere meglio perché gli assistenti virtuali con cui lavoriamo tutti i giorni, quali Siri, si limitino a rispondere a semplici domande o ad eseguire soltanto un insieme limitato di compiti.

Un altro topic balzato talvolta agli onori della cronaca, è l'insensibilità delle intelligenze artificiali alle tragedie.

La compagnia di trasporti privati Uber (Uber Technologies inc, s.d.) è stata travolta da uno scandalo nel 2014, durante la crisi degli ostaggi di Sydney (Olimpio Guido, 2014). Tra il 15 ed il 16 dicembre 2014, il fondamentalista islamico Man Haron Monis, ha tenuto in ostaggio dieci clienti e otto dipendenti di una caffetteria Lindt nell'edificio APA di Martin Place a Sydney, Australia. Tra chi è riuscito a mettersi in salvo e tra coloro che si sono visti costretti a lasciare la zona dell'incidente il più rapidamente possibile, in molti hanno pensato di utilizzare il servizio di Uber. L'AI di Uber, ignara della situazione di crisi, in seguito all'impennata di richieste in quella specifica zona, ha raddoppiato il costo delle corse (Lee, 2014). Uber è stata accusata sui social di voler monetizzare l'emergenza, approfittando della situazione di eccezionalità provocata dall'attentato. Il meccanismo che ha aumentato i prezzi, detto *surge pricing,* è basato sulla legge della domanda e dell'offerta: quando in luogo le richieste di passaggio superano la disponibilità di guidatori, l'AI di Uber alza dinamicamente i prezzi sia per incentivare altri guidatori a spostarsi in zona per riequilibrare l'offerta, sia per scoraggiare qualche passeggero e ridurre la domanda. Purtroppo, l'AI non è in grado di capire le ragioni a monte dell'aumento della richiesta. Una situazione simile a quella di

Sydney si è ripresentata nel 2016, dopo l'esplosione di un ordigno artigianale a New York (Bendinelli, 2016).

Questi ed altri fatti simili sono la conseguenza dell'incapacità delle macchine di individuare e formulare dei problemi. Le macchine sono veloci ad eseguire i calcoli ma non sono in grado di formulare dei problemi. Pertanto, le loro capacità non andranno mai spontaneamente oltre a ciò che i progettisti hanno implementato.

La creatività rimane una componente vitale per una campagna di marketing di successo. A differenza delle macchine, gli esseri umani pensano e provano emozioni, le quali spesso guidano il processo decisionale e stimolano la creatività. L'intelligenza artificiale può sicuramente aiutare a determinare quale tipo di contenuti attira maggiormente i click degli utenti, basandosi su milioni di esempi a disposizione. Ma quando si tratta di originalità e pensiero creativo, nessuna macchina può sostituire il cervello umano. L'innovazione proviene sempre da noi.

In definitiva, l'Intelligenza Artificiale diventerà sempre più efficiente, e diventerà uno strumento che ogni marketer vorrà utilizzare nella sua strategia. Rendersi conto delle sue attuali limitazioni è tuttavia indispensabile per non sviluppare eccessive aspettative sul risultato del suo utilizzo.

1.1.2 Sviluppi futuri

Negli anni a venire l'intelligenza artificiale potrebbe influenzare quasi ogni aspetto della vita quotidiana. Mentre L'AI entra nelle nostre case insieme ai nuovi prodotti high-tech, viene anche sempre più spesso adottata da enti governativi e statali per i compiti più disparati ed è oggetto di grande interesse e studio in ambito accademico.

Delle innovazioni più significative che possiamo aspettarci nel prossimo futuro, non possiamo non nominare le auto con guida autonoma. Alcuni pionieri stanno già iniziando a comparire sulle strade, con i sistemi di guida assistita, ma possiamo aspettarci che questa tecnologia progredisca notevolmente nei prossimi anni. Il Dipartimento dei Trasporti degli Stati Uniti ha iniziato a disciplinare l'uso di veicoli con trazione ad alta pressione e, di conseguenza, ha designato tre livelli di veicoli autoguidati. Attualmente è richisto ancora che il conducente umano sia al volante. L'obiettivo finale è quello di creare un'auto completamente automatizzata, la quale dovrebbe essere molto più sicura. Anche le società di logistica e i servizi di trasporto pubblico stanno cercando di incorporare la tecnologia AI per creare autotrasporti, autobus, taxi e aerei.

I campi della cibernetica e della biomedica hanno già iniziato a fondersi con l'intelligenza artificiale e questa tendenza dovrebbe continuare. Incorporando la tecnologia AI nel campo della biomedica, saremo presto in grado di migliorare il nostro corpo, dandoci maggiore forza, longevità e resistenza. Questo può essere utile nei lavori pesanti, alleggerendo il lavoro per l'umano, rendendolo sia più produttivo che meno a

rischio di incidenti e complicazioni di salute. Anche se la cibernetica può aiutarci a migliorare il nostro corpo sano, l'applicazione di questa tecnologia è davvero finalizzata ad aiutare i disabili. A quegli individui che hanno arti amputati o paralisi permanente, può essere data una qualità di vita molto più elevata. Arti cibernetici in grado di comunicare con il cervello, potrebbero diventare utili quasi quanto gli arti naturali.

Non sappiamo se sarà mai possibile creare forme di vita artificiali, anche se la fantascienza ha a lungo suggerito il concetto di robot di tipo umano capaci di interazioni complesse. Man mano che il campo della robotica progredisce e incorpora la tecnologia AI, i robot diventeranno utili in vari modi. Il già menzionato ente spaziale russo Roscosmos, sta progettando robot completamente autonomi che dovranno, in un domani non troppo lontano, occuparsi della costruzione di un avamposto umano sulla luna (Georgiou, 2018). Per quanto riguarda la realtà lavorativa di tutti i giorni, i robot potrebbero sostituirci nell'esecuzione dei lavori più pericolosi o che comportano rischi per la salute delle persone. Oggi, droni e robot meno sofisticati eseguono lavori di saldatura, anche se in genere sono controllati da un operatore tramite un telecomando. Parallelamente, gli sviluppi nella domotica mirano a renderci ogni giorno la vita più semplice.

In campo medico sono già in corso ricerche per sviluppare nuove applicazioni software che utilizzano l'AI per aiutare i medici a diagnosticare e curare i pazienti. Non ci vorrà molto tempo prima che i dispositivi indossabili, gli *smart clothes*, possano misurare costantemente i livelli di zucchero nel sangue dei diabetici e trasmettere direttamente i dati ai medici. Sono già in uso dispositivi che misurano la frequenza cardiaca, la respirazione e altre funzioni vitali. L'intelligenza artificiale può anche aiutare i pazienti a comprendere meglio le loro opzioni di cura e a comunicare in modo più efficace con gli operatori sanitari.

Nel campo dell'intrattenimento, l'intelligenza artificiale si prodiga per studiare i nostri gusti e suggerirci opere compatibili con i nostri interessi. I servizi streaming come Spotify e Netflix utilizzano l'AI per aiutarti a creare *podcast,* liste di brani che corrispondono esclusivamente alla nostra musica preferita. Ma esistono anche software in grado di guidare la creazione di contenuti, basandosi sul materiale esistente e sull'analisi delle reazioni del pubblico. In questo modo si possono realizzare sia contenuti adatti al grande pubblico, sia contenuti più mirati, qualora si individuasse una specifica nicchia di mercato.

Con la continua evoluzione dell'intelligenza artificiale, queste innovazioni saranno probabilmente messe in ombra da progressi ancora maggiori. Anche se non c'è modo di sapere con certezza fino a che punto la tecnologia AI progredirà, possiamo dare per assodato che diventerà parte integrante della nostra vita quotidiana.

2 Applicazioni di AI nel settore sales and marketing

Abbiamo utilizzato le informazioni disponibili presso Crunchbase per analizzare quali applicazioni trova oggi l'AI e come sono cambiate nel tempo a partire dalle origini. Abbiamo selezionato aziende statunitensi che appartengono alla categoria *artificial intelligence* e che hanno ricevuto più di cinque milioni di dollari in finanziamenti. Utilizzando la loro descrizione testuale e la loro data di fondazione è stato possibile ricostruire in prima approssimazione l'evoluzione delle applicazioni industriali di intelligenza artificiale.

2.1 Nascita ed evoluzione dei servizi

Allo scopo di rappresentare i processi di maturazione, applicazione e adozione di specifiche tecnologie, possiamo utilizzare uno strumento grafico detto diagramma di Gartner. Gartner è un'importante azienda nel mondo IT che si occupa di ricerche di mercato e advisoring (Gartner, 2019). Il diagramma è conosciuto anche come modello *hype cycle*, si veda Figura 1.

In sintesi, il diagramma descrive l'andamento della visibilità e della notorietà di una tecnologia in funzione delle cinque fasi principali del suo sviluppo.

La fase di innesco coincide con la fase di scoperta e diffusione mediatica della nuova tecnologia. Non esistono ancora prodotti finiti pensati per la vendita, soltanto prototipi. In generale, non sono ancora chiare quali applicazioni commerciali troverà la nuova tecnologia ed in essa investono solo dei *business angels*, convinti delle sue potenzialità e del futuro successo.

Segue la fase in cui si raggiunge il picco delle aspettative. La nuova tecnologia ottiene alcuni successi nel campo applicativo, focalizzano l'attenzione sul potenziale innovativo. Questi successi sono in realtà accompagnati da un numero molto più grande di fallimenti. Tuttavia, ci troviamo nel picco dell'entusiasmo e delle aspettative, dove i fallimenti passano inosservati o vengono attribuiti alla mancanza di esperienza degli addetti ai lavori. Ci si aspetta un superamento rapido degli ostacoli incontrati e l'imminente invasione del mercato da parte di nuovi prodotti in grado di rendere obsoleti quelli basati su tecnologie precedenti.

Inevitabilmente, segue la fase di disillusione: la nuova tecnologia non si rivela efficace come sperato ed un numero significativo dei produttori fallisce. Le startup che sopravvivono sono quelle che hanno individuato la giusta nicchia di mercato e il giusto metodo di applicazione dei loro prototipi.

Successivamente, queste startup iniziano ad accumulare esperienza e *best practice.* Diviene più chiaro qual è il settore di mercato realmente interessato ai nuovi servizi. Si

diffonde la conoscenza delle autentiche potenzialità della tecnologia e vengono proposti prodotti di seconda e terza generazione. A questo punto la tecnologia è considerata matura, viene adottata da nuovi consumatori e nascono nuovi produttori. Ripartono gli investimenti nel settore.

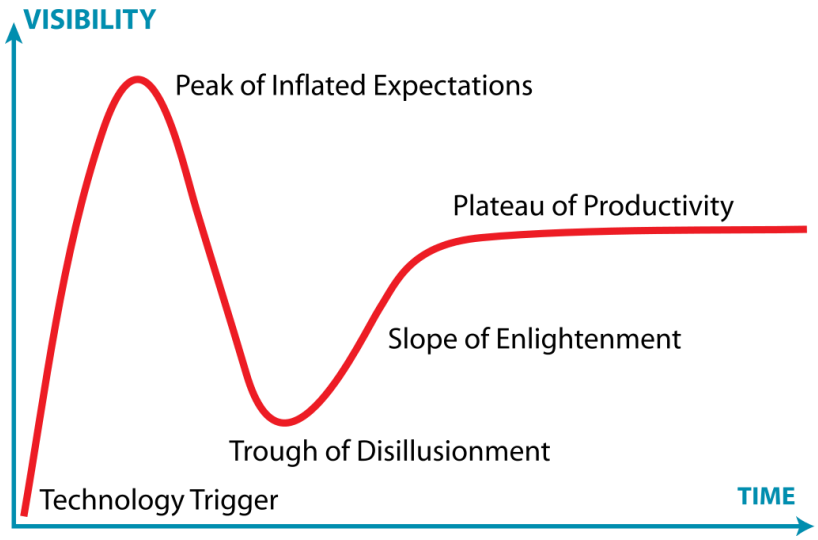


Figura 1: Diagramma di Gartner

Per quanto riguarda l'intelligenza artificiale, possiamo dedurre a che punto del grafico ci troviamo?

Abbiamo analizzato le proposte di valore delle aziende nel nostro database. In base alla loro descrizione testuale, le abbiamo suddivise per tipo di servizi offerti e ne abbiamo osservato la distribuzione temporale. Sono emerse tre principali macrocategorie di attività a cui le aziende si dedicano: *data mining, improve marketing ROI, next best action.*

Nella categoria *data mining* troviamo la maggior parte delle startup fondate nei primi anni del 2000. In quel periodo l'informatizzazione delle aziende era ormai divenuta la prassi ed iniziavano ad esistere corpose banche dati all'interno di ogni azienda.

Alcune startup nacquero insieme all'intuizione secondo la quale questi dati, se opportunamente trattati, potevano costituire una preziosa risorsa per l'azienda. Con il tempo, accortesi della validità di questa intuizione, quasi tutte le aziende si sono dotate di sistemi di *data analytics* e l'offerta di servizi si è evoluta di conseguenza. Nella seconda decade del 2000 l'opzione *make or buy* per le piattaforme dati è divenuta una realtà. Una startup neonata può scegliere di adottare una soluzione personalizzata per la gestione delle banche dati, sfruttando sia il *know-how* che l'hardware di fornitori altamente specializzati.

Altre startup si occupano invece di raccolta dati. Appurato che le informazioni hanno valore ed esaurito il potenziale dei dati prodotti internamente, un'azienda ha bisogno di raccogliere informazioni dall'esterno. Queste possono riguardare trend di mercato, andamento di borse e titoli, attività di fornitori, clienti ed anche concorrenti. Alcune startup propongono l'utilizzo di metodi innovativi basati sull'intelligenza artificiale per individuare, elaborare ed infine rivendere questi preziosi *insights.*

La seconda categoria si compone di quelle startup che si occupano dell'ottimizzazione della pubblicità, o in altre parole, di massimizzare il *return on investment (ROI)* del marketing. La maggior parte delle startup attive in questo settore sono state fondate negli anni di passaggio tra la prima e seconda decade del 2000. L'obiettivo di queste è il collegamento efficiente di domanda e offerta.

Molteplici startup nate negli anni immediatamente precedenti, si occupavano della qualità e del tipo di contenuti trasmessi negli spot pubblicitari attraverso i mezzi tradizionali. Venivano analizzati gli spot di maggior successo per individuarne gli attributi caratterizzanti. Si ottenevano così delle linee guida per la realizzazione di spot pubblicitari della massima efficacia. Anche se sicuramente sono stati ottenuti risultati importanti e grandi successi, il punto di svolta, a partire dal quale l'intelligenza artificiale ha fatto la differenza, è stato l'inizio della diffusione di massa dei dispositivi smartphone e tablet.

Questa diffusione avviene proprio nel periodo che stiamo analizzando: intorno al 2010. Grazie a nuovi mezzi di comunicazione, in grado di fungere anche da nuova sorgente di informazione abbondante e precisa, è stato possibile raffinare la ricerca e l'individuazione di segmenti di mercato ad un livello senza precedenti. Si è giunti infine al *segment-of-one marketing*. È possibile conoscere età, sesso, residenza, preferenze e molto altro di ogni singola persona. Di conseguenza sono nate startup che offrono altissima personalizzazione dei contenuti e pubblicità mirata, in modo da sfruttare al massimo tutte le risorse che le aziende investono nel marketing. Oggi, ogni azienda può scegliere di farsi pubblicità solo presso le altre aziende o i consumatori ritenuti più idonei ad acquistare i servizi o i prodotti proposti. In particolare, alcune startup propongono servizi di ricerca intelligente dei clienti, in particolare per *business to business*.

Dopo la nascita del data mining e dopo averne visto le sue prime intuitive applicazioni, negli ultimi dieci anni si sta diffondendo un nuovo tipo di servizio innovativo, che sfrutta in modo più profondo le potenzialità dell'intelligenza artificiale. Parliamo di *next-best-action analytics* o marketing contestuale (Excelle S.r.l, s.d.). Si tratta di una forma di marketing in cui le diverse azioni che si possono intraprendere sono guidate sia dagli interessi e dalle esigenze del cliente, che dagli obiettivi commerciali dell'organizzazione. Comprendiamo subito la differenza con gli approcci tradizionali, in cui prima si crea una proposta per un prodotto o servizio e solo successivamente lo si propone ai potenziali clienti. Richiede strumenti di analisi avanzati e grandi moli di dati, i quali devono anche essere coerenti, genuini e affidabili, ma in cambio permette di ottimizzare le interazioni con i clienti a tutti i livelli.

Next-best-action è un termine che viene spesso confuso con *next-best-offer*. *Next-best-offer* significa concentrarsi esclusivamente sull'offerta: l'AI sceglie accuratamente i potenziali clienti, i prodotti da offrire, magari a prezzi convenienti, ed il momento in cui inviare l'offerta. Ma questa è solo una delle possibili azioni da intraprendere. Il termine *Next-best-action* è più onnicomprensivo: include una varietà di azioni potenziali, tra cui azioni di servizio, messaggistica, coaching (assistente virtuale per il personale interno) ed anche la possibilità di determinare quando eventualmente è controindicato fare un'offerta.

Osserviamo, in Figura 2, la distribuzione temporale della fondazione delle startup. In arancione le startup che si occupano di *data analysis*, in verde quelle per l'ottimizzazione del marketing ed in azzurro *next-best-action analytics*.

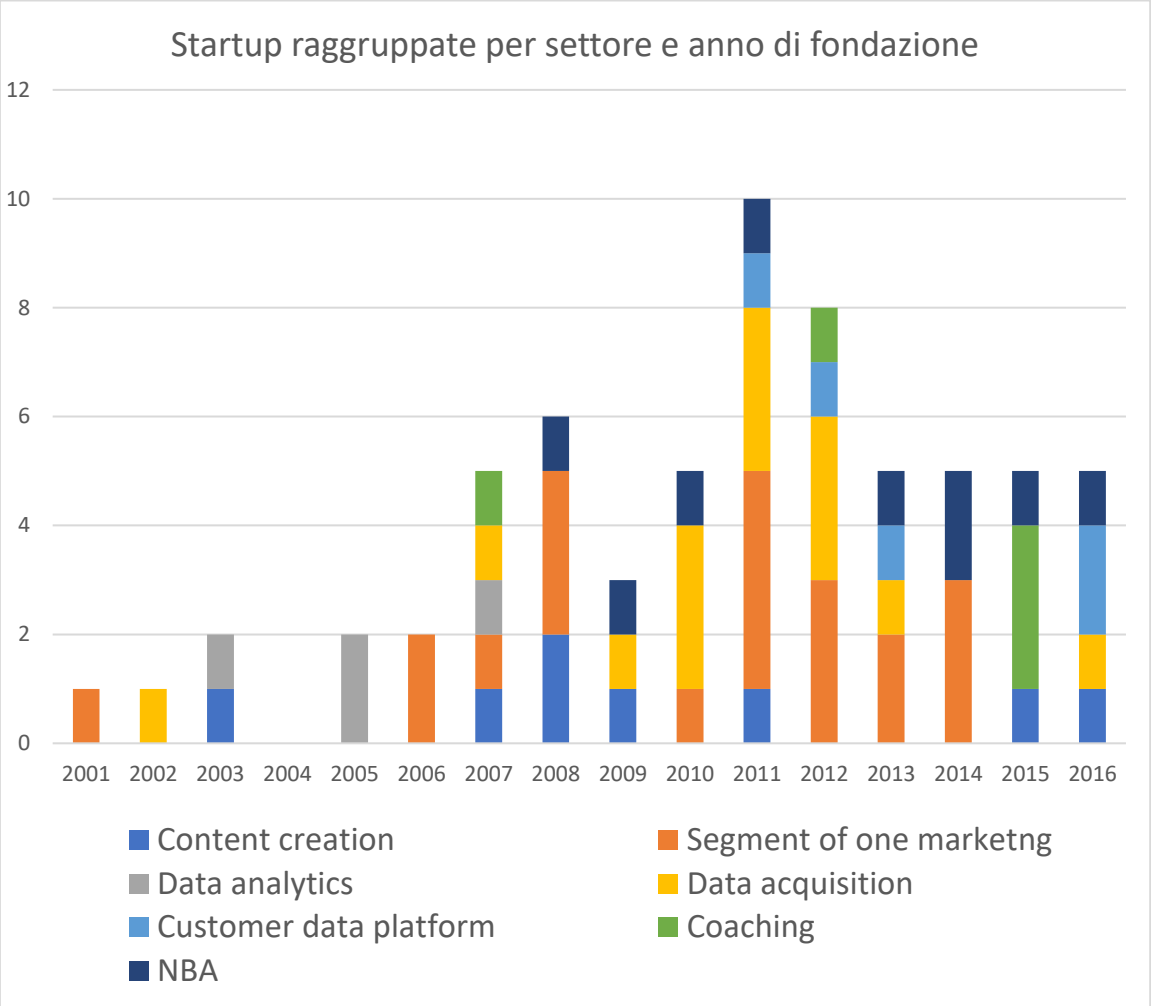


Figura 2: Numero di imprese fondate per anno, distinte per settori

La nostra analisi ci permette di concludere che, dal punto di vista del diagramma di Gartner, ci troviamo nel pieno del *plateau of productivity*. Le potenzialità e i limiti dell'intelligenza artificiale sono ormai noti e possiamo individuare almeno diverse generazioni di prodotti.

Vediamo come i servizi di puro *data analytics* siano in un certo qual modo evoluti: dall'analisi dei dati interni di singole aziende (in giallo) alle *customer data platform* (in rosso), che raccolgono dati di molteplici società e permettono una gestione agile, evitando costosi investimenti in hardware. Ad oggi, i servizi diventano giorno per giorno accessibili a realtà sempre più piccole, grazie allo sviluppo di tecnologie parallele quali il *cloud computing*. Una startup giovane può acquistare una suite di servizi accessibili semplicemente via browser. Tramite essi, può avere accesso alla stessa rete di

informazioni e contatti utilizzati e condivisi sia da altre startup che da affermate multinazionali.

La personalizzazione dei contenuti è distribuita quasi uniformemente. Segmentare il mercato è un concetto economico assodato ed il *segment-of-one* non è che la sua massima espressione. Possiamo argomentare che l'intelligenza artificiale si stata studiata e sviluppata avendo anche in mente questo obiettivo, per esempio da colossi quali Google. I servizi di *NBA*, invece, sono di concezione più recente. Sono un prodotto più avanzato, frutto di una conoscenza più approfondita della tecnologia e dei suoi limiti.

Appunto per la presenza di questi prodotti avanzati, possiamo affermare che l'intelligenza artificiale, se non ha già raggiunto il *plateau of productivity,* si trovi quantomeno nello *slope of enlightenment.*

2.1.1 Trend attuali e startup innovative

In questa sezione esploreremo lo stato dell'arte in termini di offerta di servizi e di proposte innovative, correlate e non alle tre macrocategorie discusse in precedenza.

La gestione dei dati personali e la privacy sono temi di grande attualità. Gli utenti di internet producono ogni giorno, quasi inconsapevolmente, enormi quantità di dati molto utili per fare analisi di mercato e advertising. Questi dati, oltre ad essere oggetto di innumerevoli discussioni e problematiche di sicurezza, possono essere, in certe misure e in accordo alle leggi vigenti, utilizzati da compagnie *for profit*.

I dati non sono tutti uguali e di per sé non generano automaticamente valore. Ai dati deve essere attribuito un significato, devono dare risposte a domande. Sperabilmente, all'acquisizione di nuovi dati deve corrispondere l'acquisizione di nuova conoscenza utile a impostare nuove strategie di vendita o di produzione. Si devono trasformare i dati in nuovi asset per l'azienda. Quindi una startup che volesse generare una proposta di valore dovrà individuare una sorgente di dati innovativa oppure riutilizzare una risorsa nota in modi completamente nuovi. Le startup offrono sistemi di raccolta e analisi di dati che permettono alle aziende di espandere il proprio spettro d'azione ed aumentare l'efficacia delle loro azioni. I dati utilizzati sono estremamente eterogenei e variegati. Tra gli approcci più comuni, e al contempo complessi troviamo quelli legati all'analisi di dati provenienti dai social media. Si tratta principalmente di file testuali e multimediali. Gli approcci possibili a questo tipo di dati sono infiniti. Con l'intelligenza artificiale si possono utilizzare le immagini ed i video, ad esempio, per individuare i trend e i topic più gettonati del momento, allo scopo di guidare la *content creation.* Ma si possono adottare approcci completamente diversi: la BrandTotal (BrandTotal, s.d.), startup americana fondata nel 2016, raccoglie dati di migliaia di aziende per offrire ai propri clienti una *competitor analysis*. Sapere cosa fanno e come si muovono i diretti concorrenti può essere un vantaggio decisivo per qualsiasi azienda. Questa intuizione sta guidando verso il successo BrandTotal, la quale è cresciuta rapidamente raccogliendo 8 milioni di dollari in investimenti in soli due anni.

Quando i dati sono divenuti disponibili, due delle prime e più fruttuose applicazioni di intelligenza artificiale sono state la pubblicità mirata e la personalizzazione dei contenuti. Molte startup si dedicano con successo alla profilazione degli utenti per poter poi offrire ai loro clienti una forma di advertising estremamente efficace. In termini di *value proposition* i servizi offerti da queste startup permettono, come già menzionato, di massimizzare il *ROI* nel settore marketing.

Un altro tipo di servizio innovativo introdotto è il cosiddetto *driven product discovery.* Si cattura l'attenzione del potenziale cliente su smartphone o pc tramite un gioco oppure un quiz. L'AI di *Qzzr* (Qzzr, s.d.), startup fondata nel 2010, realizza dei quiz su misura per l'utente che naviga in rete. All'utente vengono sottoposte domande su argomenti (stimati essere) di suo interesse, di difficoltà crescente. Lo scopo è guidarlo, tramite l'interazione attiva, alla scoperta di un nuovo servizio o prodotto. Questo approccio si è rivelato più efficace dei tradizionali metodi di pubblicità. Sottoporre ad una sfida le conoscenze e competenze dell'utente, ha lo scopo di scoprirne i limiti e di suggerire un modo, un prodotto o un servizio, per superarli.

Infine, negli ultimi anni le figure degli *influencer* sono diventate una realtà economica di grande rilevanza: il rapporto di fiducia che essi costruiscono con il loro pubblico ha un valore intrinseco elevatissimo. L'*influencer marketing* è un canale di marketing online che sta vivendo una fase di crescita rapidissima, in competizione e a spesso anche soverchiando altre strategie affermate come *e-mail marketing, advertisement display* e *paid search.*

La startup Influential (Influential, s.d.) utilizza il suo network di più di 300000 influencer per vendere pubblicità sui social media. Ha ricevuto più di 36 milioni in investimenti ed è in partnership con svariate *Top 500 Fortunes* quali McDonald's, Toyota e Samsung.

Chi invece volesse intraprendere la carriera di *influencer,* può contare sull'aiuto dell'AI sviluppata da Post Intelligence (Post Intelligence - Buy Automatic Instagram Likes & View, s.d.). Post Intelligence fornisce un servizio di *Social Media Assistant*, atto a migliorare la quantità dei contenuti postati su Instagram ed acquisire follower. L'offerta prevede un pacchetto minimo di follower e condivisioni dei post. È risaputo, tuttavia, che acquistare like e follower è inutile in quanto le interazioni provengono spesso da profili fasulli creati ad-hoc. Quindi non rispecchiano interessi reali di persone verso influencer o di clienti verso un brand. Post Intelligence distribuisce strategicamente le interazioni social con i sui clienti nel tempo. In questo modo gli algoritmi di Instagram vengono indotti a mantenere altamente visibili i post in modo che questi vengano visualizzati dagli utenti reali. Si innesca così il circolo virtuoso per cui non è più necessaria alcuna spinta alla visibilità. Post Intelligence ha raccolto più di 11 milioni di dollari in finanziamenti.

Nel ramo *business to business,* un tipico esempio di servizio basato sull'intelligenza artificiale sono i *chat bot.* Sono diffusi ormai da molti anni e spesso, navigando in Internet, nelle home page di vari siti gli utenti vedono aprirsi delle finestre di chat in cui L'AI da loro il benvenuto e li informa di essere a loro disposizione per qualsiasi informazione. Un'evoluzione più recente è l'assistente virtuale vocale. Gli assistenti virtuali non sono ancora in grado di sostituire gli operatori umani ai call center ma sono

comunque multo utili: possono rispondere alle domande più elementari dei clienti e svolgere alcune operazioni semplici. Se non sono in grado di comprendere a fondo ciò che il cliente vuole, cercano comunque di metterlo in comunicazione con l'operatore più opportuno. I call center registrano ogni giorno centinaia, se non migliaia o più, telefonate ogni giorno. Molte di queste vengono selezionate come campioni per allenare le intelligenze artificiali dei chatbot. Vedremo più avanti perché non si utilizzano semplicemente tutte le informazioni disponibili.

Startup di fondazione più recente, si occupano di ricerca intelligente di partner e clienti nel ramo industriale, per connettere efficacemente domanda e offerta e costruire relazioni mutuamente profittevoli e durature nel tempo. Si parla in questo caso di *customer lifetime value management* attraverso l'ottimizzazione del marketing e delle interazioni. L'intelligenza artificiale assiste le aziende in tutte le iterazioni coi clienti, non solo nella fase di acquisizione. Per questo si parla di *lifetime value*. Entriamo nel campo delle *next best actions* e del *coaching*. Startup di nuova generazione forniscono servizi di assistenza intelligente.

Il *coaching* è una delle applicazioni di maggior successo. Sia singoli professionisti che operatori di call center, sono assistiti da un'AI che ascolta ed analizza le conversazioni dal vivo e fornisce suggerimenti in tempo reale. Un esempio di startup che fornisce tipo è *Chorus.ai* (Conversation Intelligence for Sales Teams, s.d.). L'AI di Chorus ascolta ed analizza le conversazioni dei *sales team.* Studia sia i successi, per capire le strategie vincenti dei venditori migliori, che i fallimenti, per capire quali sono i passi falsi da evitare. In questo modo, in caso di situazioni nuove o impreviste, l'AI è anche in grado di beneficiare dell'intuizione e della capacità di *problem solving* degli operatori umani. Così può autoaggiornarsi ed imparare sempre nuove tecniche e strategie, da suggerire prontamente.

3 Dataset

3.1 Crunchbase

Figura 3: Crunchbase, logo ufficiale. © 2019 Crunchbase Inc.

> *Crunchbase is the leading platform for professionals to discover innovative companies, connect with the people behind them, and pursue new opportunities. Over 55 million professionals—including entrepreneurs, investors, market researchers, and salespeople—trust Crunchbase to inform their business decisions. And companies all over the world rely on us to power their applications, making over a billion calls to our API each year* (Crunchbase Inc).

3.1.1 Storia

Crunchbase è un *open dataset* che raccoglie informazioni su startup, investimenti, tendenze, investitori, fondatori, individui in ruoli di leadership, fusioni, acquisizioni ed altre informazioni correlate. Queste informazioni sono sottomesse su base volontaria dai soggetti interessati, e vengono pubblicate previa validazione e revisione da parte dei moderatori del sito.

Nasce nel 2007 col nome CrunchBase: realtà sussidiaria della piattaforma TechCrunch (TechCrunch – Startup and Technology News), atta alla raccolta di informazioni sulle startup che venivano menzionate negli articoli pubblicati sulla piattaforma principale. Diviene indipendente nel 2015, cambiando il suo nome da CrunchBase in Crunchbase.

L'informazione raccolta da Crunchbase è un genere di risorsa che si rivolge in primis ai venture capitalist. Invece, dal punto di vista di un'azienda, avere un profilo Crunchbase comporta il vantaggio di fornire a possibili azionisti (come anche ad investitori, partner, clienti e potenziali acquirenti) un ulteriore punto di dati sui vostri prodotti e servizi. Per questo sono sempre di più le diverse aziende che aprono un loro profilo. Un altro vantaggio è la possibilità di collegare al proprio profilo altre risorse importanti, quali news che descrivono l'azienda, l'organizzazione o gli individui. Proprio come in un social media, vengono anche notificate le attività di altre aziende seguite o di interesse.

I profili non sono esclusivamente aziendali. Singoli professionisti o team possono utilizzare la piattaforma per contattare persone ed aziende interessate alla loro attività.

Oltre ai profili gratuiti, Crunchbase dispone di una suite di prodotti che garantiscono diversi livelli di accesso al database.

Secondo le informazioni pubblicate nel suo stesso dataset (Crunchbase inc), dopo la separazione da TechCrunch, Crunchbase ha raccolto 26.5 milioni di dollari da 8 diversi investitori in 3 round di finanziamenti, tra il 2015 e il 2017.

3.1.2 Crunchbase per la ricerca

Crunchbase consente ai ricercatori, caso per caso, accesso completo o parziale al database. Si tratta di una sorgente di dati molto giovane e con molto potenziale ancora da esplorare. È interessante comprendere le potenzialità di questo innovativo database per la ricerca economica e manageriale. Crunchbase è un database *crowd-sources*, ovvero costruito su base volontaria, nutrito ed aggiornato dagli utenti stessi. Pertanto, le sue percentuali di copertura e di rappresentazione della realtà non sono chiaramente definite. La sua portata varia a seconda dei paesi e dei settori economici in esame. Ricordiamo che nasce da *TechCrunch*, il quale si focalizza su aziende operanti nel *tech*, appunto. In ogni caso, le statistiche aggregate per paese e per anno sui finanziamenti di capitale di rischio tendono ad essere ragionevolmente simili alle cifre riportate da fonti alternative più consolidate. Tra queste annoveriamo KickStarter (Kaminski, 2016), Twitter (Tata, 2017), e LinkedIn (Nuscheler, 2016).

3.1.3 Reperimento delle informazioni.

L'ecosistema di Crunchbase è basato sul *crowdsourcing:* è un progetto sviluppato collettivamente da parte dell'azienda ideatrice e da persone esterne. Collabora infatti con più di 3.500 società d'investimento in tutto il mondo, le quali inviano costantemente alla piattaforma gli aggiornamenti dei loro portafogli. Questo garantisce a Crunchbase accesso diretto a informazioni sempre aggiornate. In cambio, Crunchbase permette loro accesso alle preziose informazioni contenute nel database. Inoltre, nel sito è attiva una community di dirigenti, imprenditori e investitori, i quali contribuiscono attivamente alla crescita del database, creando e aggiornando le pagine dei profili aziendali.

Crunchbase dispone di un team interno di *data scientist* che monitorano il database e convalidano l'accuratezza dei dati. Gli analisti di Crunchbase sono gli esperti che forniscono la validazione manuale dei dati ed il loro mantenimento. Analizzano anche le principali interconnessioni dei dati per ricavare insights e sviluppare algoritmi di apprendimento automatico. Sono assistiti nel loro compito dall'intelligenza artificiale, la quale rileva anomalie e conflitti nei dati.

3.2 Descrizione dei campioni

Il dataset da noi utilizzato consiste in sette file in formato CSV (Shafranovich, 2005) estratti da Crunchbase il giorno 06-03-2019. Il dataset è stato ottenuto selezionando tutte le aziende operanti nel settore *sales and marketing*.

CSV (comma-separated variables) è un formato file utilizzato nei fogli elettronici e nei database in cui si utilizza il carattere virgola ‘,’ (in inglese comma, da cui il nome) per separare i valori contenuti in ogni riga. Il file di testo assume quindi una formattazione tabulare riconoscibile dai fogli elettronici e dai database. I fogli di calcolo quali Microsoft Excel permettono di suddividere automaticamente i dati in colonne utilizzando la virgola come delimitatore. Un file CSV può contenere una sola tabella. Il dataset a nostra disposizione è quindi costituito da sette tabelle. Andremo ora a descriverle nei contenuti, specificando, colonna per colonna, da che tipo di dati sono composte e come sono rappresentati.

1- Acquisitions

Questo file contiene 5982 entry, ognuna delle quali rappresenta l’acquisizione di un’azienda da parte di un’altra. Vengono fornite le indicazioni geografiche delle organizzazioni acquirente ed acquisita, il tipo, la data ed il prezzo di acquisizione. Le colonne della tabella sono riportate nella pagina successiva in Tabella 2.

2- Category_groups

Questo file contiene 680 righe, ognuna delle quali rappresenta una categoria. Ogni organizzazione è associata ad una o più categorie sulla base delle caratteristiche delle attività economiche. A loro volta le categorie sono associate a dei raggruppamenti detti *category grouplist*, traducibile con macrocategorie, i quali accorpano le categorie simili tra loro. Le macrocategorie sono 46.

Nome colonna	Tipo di variabile	Significato
category_uuid	Nominale	Codice identificativo della categoria
category_name	Categorica	Nome della categoria
category_groupList	Categorica	Macrocategorie associate

Tabella 1: Category_groups

Nome colonna	Tipo di variabile	Significato
acquiree_country_code	Categorica	Codice identificativo della nazione di appartenenza dell'organizzazione acquisita
acquiree_state_code	Categorica	Codice identificativo dello stato di appartenenza dell'organizzazione acquisita
acquiree_region	Categorica	Regione di appartenenza dell'organizzazione acquisita
acquiree_city	Categorica	Città di appartenenza della sede principale dell'organizzazione acquisita
acquirer_country_code	Categorica	Codice identificativo della nazione di appartenenza dell'organizzazione acquirente
acquirer_state_code	Categorica	Codice identificativo dello stato di appartenenza dell'organizzazione acquirente
acquirer_region	Categorica	Regione di appartenenza dell'organizzazione acquirente
acquirer_city	Categorica	Città di appartenenza della sede principale dell'organizzazione acquirente
acquisition_type	Categorica	Specifica la tipologia di acquisizione, i possibili valori sono: "*acquihire", "acquisition", "lbo", "management_buyout", "merge"*.
acquired_on	Data	Data in formato gg/mm/aaaa dell'acquisizione
price_usd	Ordinale	Prezzo dell'acquisizione espresso in USD
price	Ordinale	Prezzo espresso nella valuta in cui è stata eseguita l'acquisizione, se diverso da USD
price_currency_code	Categorica	Codice identificativo della valuta in cui è stata eseguita l'acquisizione, se diverso da USD
acquiree_uuid	Nominale	Codice identificativo dell'organizzazione acquisita
acquirer_uuid	Nominale	Codice identificativo dell'organizzazione acquirente
acquisition_uuid	Nominale	Codice identificativo dell'acquisizione
created_at	Data	Data di creazione della entry nel database Crunchbase
updated_at	Data	Data di ultimo aggiornamento della entry nel database Crunchbase

Tabella 2: Acquisitions

3- Funding Rounds

Questo file descrive 21480 funding rounds, specificandone la data, quanto denaro è stato investito (in USD), da quanti e quali investitori e fornendo una *post money evaluation* (Majaski, 2019) per l'organizzazione che ha ricevuto il finanziamento.

Nome colonna	Tipo di variabile	Significato
fundingRound_ investmentType	Categorica	Specifica di che tipo di investimento si tratta
fundingRound_announcedOn	Data	Data di annuncio del funding round, espressa in formato gg/mm/aaaa
fundingRound_ annoncedOnYear	Ordinale	Anno di annuncio del funding round
fundingRound_ raisedAmountUsd	Ordinale	Totale di denaro raccolto espresso in USD
fundingRound_ postMoneyValuationUsd	Ordinale	Post money evaluation dell'organizzazione dopo il funding round, espresso in USD
fundingRound_investorCount	Ordinale	Numero di investitori che hanno partecipato al round
fundingRound_uuid	Nominale	Codice identificativo unico del funding round
fundingRound_companyUuid	Nominale	Codice identificativo unico dell'organizzazione ricevente
fundingRound_investorUuids	Nominale	Lista di codici identificativi unici degli investitori
fundingRound_createdAt	Data	Data di creazione della entry nel database Crunchbase
fundingRound_updatedAt	Data	Data di ultimo aggiornamento della entry nel database Crunchbase

Tabella 3: Funding Rounds

4- Companies

Questo è il principale file con cui si è lavorato nella realizzazione di questa tesi. Questo sottoinsieme di 62199 imprese contiene, per ognuna, informazioni su data e luogo di fondazione, categorie economiche di appartenenza, investimenti ricevuti, numero di dipendenti. Infine, ci dice se l'azienda è attualmente *"operating"* oppure se ha fatto una exit, se è stata acquisita o se ha chiuso.

Nome colonna	Tipo di variabile	Significato
organization_type	Categorica	Viene dato il valore *"organization"*
organization_roles	Categorica	Ruoli della compagnia.
organization_primaryRole	Categorica	Viene dato il valore *"company"*
organization_countryCode	Categorica	Codice identificativo della nazione di appartenenza dell'organizzazione
organization_stateCode	Categorica	Codice identificativo dello stato di appartenenza dell'organizzazione
organization_region	Categorica	Regione di appartenenza dell'organizzazione
organization_city	Categorica	Città di appartenenza della sede principale dell'organizzazione
organization_status	Categorica	Stato di operatività.
organization_foundedOn	Data	Data di fondazione dell'organizzazione, espressa in formato gg/mm/aaaa
organization_foundedOnYear	Ordinale	Anno di fondazione dell'organizzazione
organization_closedOn	Data	Data di chiusura, espressa in formato gg/mm/aaaa
organization_categoryList	Categorica	Lista di categorie economiche che descrivono le attività dell'organizzazione.
organization_categoryGroupList	Categorica	Lista delle macrocategorie associate all'organizzazione
organization_fundingRounds	Ordinale	Numero di investimenti ricevuti dall'organizzazione
organization_fundingTotalUsd	Ordinale	Totale del denaro raccolto tramite investimenti dall'organizzazione, espresso in USD
organization_lastFundingOn	Data	Data dell'ultimo funding round ricevuto dall'azienda, espressa in formato gg/mm/aaaa
organization_employeeCount	Intervallo	Numero di dipendenti dell'organizzazione.
organization_uuid	Nominale	Codice identificativo unico dell'organizzazione
organization_createdAt	Data	Data di creazione della entry nel database Crunchbase
organization_updatedAt	Data	Data di ultimo aggiornamento della entry nel database Crunchbase
organization_exit	Categorica	Valore *true* se l'organizzazione ha eseguito una exit, *false* altrimenti

Tabella 4: Companies

5- Investors

Questo file descrive 11526 investitori, tra privati ed organizzazioni. Si danno indicazioni geografiche e sull'anno di fondazione, informazioni sul numero di investimenti e sul capitale totale investito.

Nome colonna	Tipo di variabile	Significato
investor_type	Categorica	Specifica di che tipo di investitore si tratta
investor_roles	Categorica	Ruolo o ruoli coperti dall'investitore
investor_countryCode	Categorica	Codice identificativo della nazione di appartenenza dell'investitore
investor_stateCode	Categorica	Codice identificativo dello stato di appartenenza dell'investitore
investor_region	Categorica	Regione di appartenenza dell'investitore
investor_city	Categorica	Città di appartenenza dell'investitore
investor_foundedOn	Data	Data di fondazione dell'investitore, se questi è una compagnia, espressa in formato gg/mm/aaaa
investor_founded_on_year	Ordinale	Anno di fondazione
investor_closedOn	Data	Data di chiusura, qualora l'investitore fosse una compagnia e questa avesse chiuso
investor_investmentsCount	Ordinale	Totale di investimenti effettuati dal singolo investitore
investor_totalFundingUsd	Ordinale	Totale di denaro investito dall'investitore, espresso in USD
investor_uuid	Nominale	Codice identificativo unico dell'investitore
investor_updatedAt	Data	Data di ultimo aggiornamento della entry nel database Crunchbase

Tabella 5: Investors

6- Investments

Questo file da 34327 entry, ci dice per ogni funding round e per ogni investitore coinvolto in esso, se questi ne è il *lead investor* (BusinessDictionary.com, 2019). Questo file potrebbe essere eliminato senza perdita di informazione semplicemente aggiungendo al file precedente una colonna *fundingRound_leadInvestor* contenente l'uuid del lead investor.

Nome colonna	Tipo di variabile	Significato
investment_ fundingRoundUuid	Nominale	Codice identificativo unico del funding round
investment_investorUuid	Nominale	Codice identificativo unico dell'investitore
investment_isLeadInvestor	Categorica	Valore booleano *"t"* o *"f"*, vale *true* se l'investitore è anche il lead investor, *false* altrimenti

Tabella 6: Investments

7- Ipos

L'ultimo file (Tabella 7) contiene informazioni aggiuntive sulle 728 organizzazioni del nostro database che hanno fatto una IPO (Treccani.it, 2019). Un' *Initial Public Offering* (IPO) è il processo che trasforma un'impresa privata in una società pubblica, le cui azioni sono quotate in borsa. A seguito dell'IPO, una società diviene di proprietà degli azionisti che acquistano le sue azioni. In questa tabella sono indicati la data dell'IPO, il simbolo di *stock* (Hayes, 2019) dell'organizzazione, il nome e la sede dello *stock exchange* (Wikipedia, 2019) ovvero la borsa valori in cui è avvenuta l'IPO. È indicato l'*opening price* (Fedorov, 2019) totale delle azioni vendute in USD e in valuta locale, ed il totale del denaro raccolto in USD.

Nome colonna	Tipo di variabile	Significato
country_code	Stringa	Codice identificativo della nazione di appartenenza dell'organizzazione
company_state_code	Stringa	Codice identificativo dello stato di appartenenza dell'organizzazione
region	Stringa	Regione di appartenenza dell'organizzazione
city	Stringa	Città di appartenenza della sede principale dell'organizzazione
stock_exchange_symbol	Stringa	Borsa valori in cui è avvenuta l'IPO
stock_symbol	Categorica	Simbolo di stock dell'organizzazione
went_public_on	Data	Data dell'IPO espressa in formato gg/mm/aaaa
price_usd	Ordinale	Opening price totale espresso in USD
price	Ordinale	Opening price totale espresso in valuta locale
price_currency_code	Categorica	Codice identificativo della valuta in cui è stata eseguita l'IPO, se diverso da USD
money_raised_usd	Ordinale	Denaro raccolto a seguito dell'IPO
ipo_uuid	Nominale	Codice identificativo univoco della IPO
company_uuid	Nominale	Codice identificativo univoco dell'organizzazione
created_at	Data	Data di creazione della entry nel database Crunchbase
updated_at	Data	Data di ultimo aggiornamento della entry nel database Crunchbase

Tabella 7: IPOS

4 Preprocessing dei dati

Elaboreremo i file CVS in modo da trasformarli in una o più matrici di numeri. Ogni riga costituisce un campione mentre le colonne rappresentano *features,* caratteristiche, che lo descrivono. A seconda del task, rimuoveremo le colonne inutili o "rumorose" per lasciare soltanto colonne portatrici di informazione utile. A quel punto la matrice potrà essere utilizzata come *training set* per un algoritmo di learning. Non sempre le colonne utili sono già disponibili, vedremo come sarà necessario combinare le informazioni in nostro possesso per produrne di nuove più significative.

Il linguaggio di programmazione ad alto livello Python (Python Software Foundation (US), 2019) dispone di un vasto insieme di API per interagire con file CSV. In questo elaborato si è utilizzata la libreria pandas (pydata.org, 2019) per importare ed elaborare i dati in modo da lavorare con oggetti tabellari equivalenti ad un database. Molto spesso le righe delle tabelle non contengono un valore valido per ogni colonna. In questi casi è presente il valore di default *"NA"*. Invocando il metodo *info()* della libreria *pandas,* otteniamo il conteggio degli elementi non nulli per ogni colonna della tabella. La libreria missingno (ResidentMario, 2019) fornisce un utile strumento per la visualizzazione dei valori *"NA"*. In Figura 4 vediamo l'output del metodo *info()* sulla tabella companies_MKT_anonym.csv. Questo file sarà il principale oggetto dei nostri esperimenti. In Figura 5, invece, vediamo rappresentata graficamente, tramite il metodo *matrix()* della libreria missingno, la distribuzione dei valori mancanti nella tabella.

Il pre-processing dei dati consisterà di tre fasi:

1) Selezione: si individuerà un sottoinsieme dei dati ritenuto significativo ed adatto ad essere analizzato con algoritmi di machine learning.
2) Cleaning: dal dataset risultante saranno rimosse tutte le informazioni ridondanti ed i valori mancanti.
3) Arricchimento: saranno aggiunte nuove variabili a *companies*, ottenute aggregando dati provenienti da altre tabelle.

```
companies_MKT_anonym.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 62199 entries, 0 to 62198
Data columns (total 21 columns):
Type                62199 non-null object
Roles               62127 non-null object
Primary Role        62199 non-null object
Country             46387 non-null object
State               25308 non-null object
Region              37061 non-null object
City                46403 non-null object
Status              62199 non-null object
Founded on          62199 non-null object
Foundation year     50759 non-null float64
Closed on           946 non-null datetime64[ns]
Cat List            62199 non-null object
Cat GruopList       62199 non-null object
Funding rounds      62199 non-null int32
Funding total $     8384 non-null float64
Last funding on     10941 non-null datetime64[ns]
Employees           50375 non-null object
uuid                62199 non-null object
Created             62199 non-null datetime64[ns]
Updated             62199 non-null datetime64[ns]
Exit                62199 non-null bool
dtypes: bool(1), datetime64[ns](4), float64(2), int32(1), object(13)
memory usage: 9.3+ MB
```

Figura 4: output del metodo info() della libreria pandas

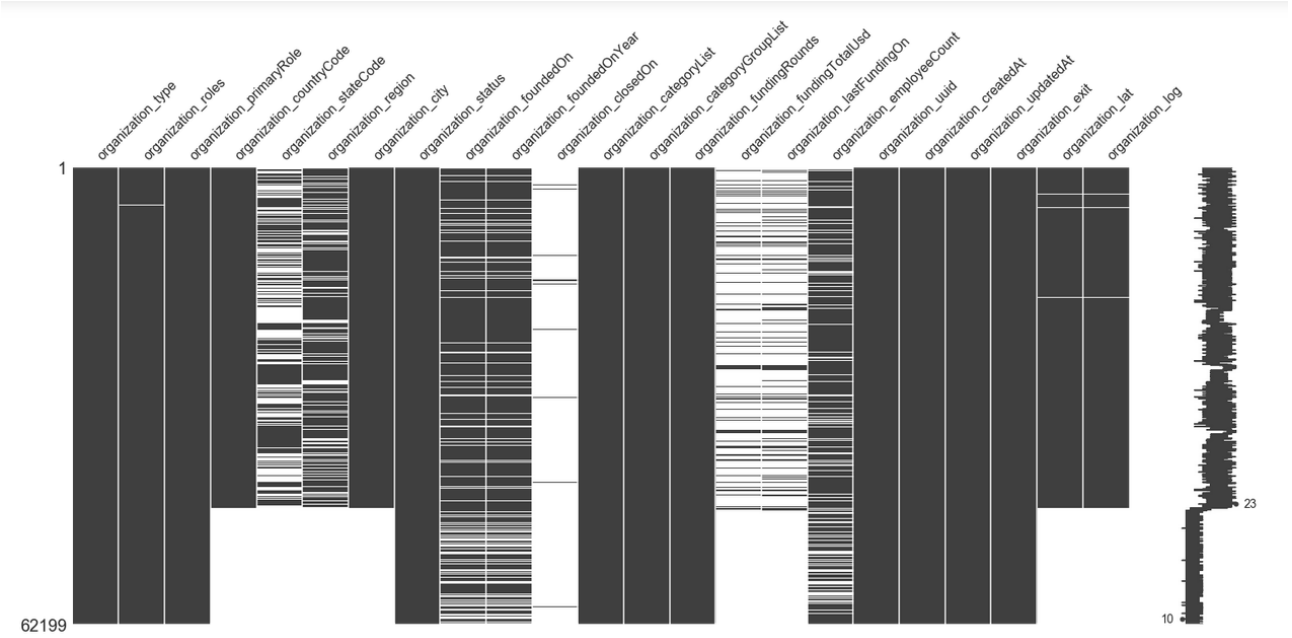


Figura 5: output del metodo matrix() della libreria missingno

Per ottenere delle variabili numeriche, il primo problema da affrontare è la conversione dei tipi di dato. Moltissime colonne sono variabili categoriche, espresse da stringhe (le quali in python sono considerate come *object*), pertanto dovremo scegliere delle appropriate funzioni per mappare queste stringhe in numeri in formato *NumPy float64* (community, 2017). Per importare il dataset utilizzeremo il metodo *read_csv* (pandas.read_csv - pandas 0.25.1 documentation, 2019) della libreria pandas. Questo metodo cerca di identificare autonomamente il tipo di dato delle colonne e distingue i valori nulli, *"NA",* i numeri e le stringhe di caratteri. Ci permette di selezionare quali colonne importare, eventualmente rinominandole, e di specificare il tipo di dato *dtype* (data type) in cui interpretare ogni colonna. Per le date in particolare è disponibile il tipo di dato *datetime* (Python Software Foundation, 2019) tuttavia non sarà utilizzato in questo elaborato. Costruiremo una funzione ad-hoc per gestire associare il concetto di data a quello di intervallo di tempo.

I campi quali *organization_categoryList* e *organization_employeeCount* sono detti categorici o nominali. Essi possono assumere solo un insieme finito di valori, espressi sotto forma di stringa. Vanno dunque convertiti in un formato utilizzabile dagli algoritmi di machine learning.

La notazione intuitivamente più immediata è detta *integer encoding* in cui ad ogni valore viene assegnato un numero intero, in modo simile ad un dizionario di coppie chiave-valore. Questa notazione è semplice da implementare e sostituisce efficacemente un valore stringa con un vettore unidimensionale. Tuttavia, presenta una grave controindicazione. Introduce infatti un'informazione fittizia e generalmente non coerente. Per i numeri interi sussiste una relazione d'ordine (possono essere maggiori minori o uguali) ed esiste un concetto di distanza. Un algoritmo di machine learning potrebbe interpretare valori quali 1, 2 e 3 come più "simili" e "vicini" rispetto a 10, 50, 100. Tuttavia, è altamente improbabile che la stessa relazione sussista tra le variabili categoriche che questi numeri identificano. Per risolvere questo problema si utilizza una notazione alternativa detta *one hot vector*. Uno *one hot vector* è un vettore trasposto di variabili *dummy*. la cui dimensione è pari alla cardinalità dell'insieme di variabili categoriche, ad esempio 46 per *category_groupList*. Ogni variabile *dummy* è associata ad uno o più valori categorici, quindi data una entry nel dataset, lo *one hot vector* associato sarà un vettore di zeri eccetto per valori 1 nelle posizioni corrispondenti ai valori categorici presenti nella entry.

Sostituire colonne del dataset con questi vettori comporta un aumento della dimensione *m* della matrice. Intuitivamente, aumentare la dimensione significa aumentare la complessità del modello e con essa il tempo di esecuzione dell'algoritmo e il numero di campioni necessari all'apprendimento. Tuttavia, l'insieme degli *one hot vector* associati ad un campo del database, è di fatto una matrice sparsa di 0 e 1, per la gestione della quale gli algoritmi di Python dispongono di subroutine altamente ottimizzate. Inoltre, dal punto di vista dell'algoritmo, i problemi relativi alla relazione d'ordine vengono completamente risolti da questa notazione. In effetti, per i loro vantaggi gli *one hot vector* trovano ampio uso nel campo del machine learning. Implementeremo dunque una funzione adatta alle nostre esigenze per la conversione delle stringhe in vettori.

4.1 Selezione delle feature

Iniziamo dunque il preprocessing del dataset per il nostro esperimento. Utilizzeremo il file *companies_MKT_anonym.csv*, a cui d'ora in poi ci riferiremo con il nome abbreviato *companies.*

In fase di selezione, dobbiamo decidere quali colonne della matrice utilizzare per i nostri algoritmi e come gestire i valori mancanti o *"NA"*. Questo secondo punto è molto delicato e sarà discusso in seguito. Iniziamo selezionando per esclusione le colonne.

È immediata l'esclusione delle seguenti colonne: *uuid*, *updated_at, created_at,* in quanto contengono informazioni significative solo per il database Crunchbase. Elementare anche l'eliminazione delle colonne *type* e *primaryRole,* le quali contengono lo stesso valore ripetuto per tutte le entry. La colonna *roles* contiene per quasi tutte le entry il valore *"company"*, per circa un migliaio il valore "*company*|*investor*", per due entry il valore "*company*|*school*|*investor*", mentre *"investor", "school*|*investor", "school*|*company", "company*|*school"*, compaiono ognuno soltanto una volta. Poiché, anche intuitivamente, non sono portatori di informazione particolarmente significativa, scegliamo di eliminare questa colonna. L'informazione contenuta nella colonna *status* rende la colonna *exit* ridondante: ai valori *"acquired"* e *"ipo"* è associato il valore *"t"* (true) mentre ad *"operating"* e *"closed"* il valore *"f"* (false). Per questo possiamo eliminare *exit* senza perdita di informazione.

Vediamo dalla Figura 5 come la colonna *closedOn* sia quasi del tutto vuota. Dalla Figura 4 vediamo che solo 946 entry hanno valori non nulli. L'informazione che portano ci sarà utile in altri ambiti, per ora ci possiamo accontentare ancora una volta della colonna *status* per sapere quali organizzazioni hanno chiuso.

La colonna *foundedOn* ci fornisce una data in formato gg/mm/aaaa. Possiamo accontentarci, in prima approssimazione, della colonna *foundedOnYear*, che contiene solo l'anno di fondazione, è semplice da gestire, in quanto già in formato numerico, ed inoltre non contiene valori *"NA"*, si veda ancora una volta Figura 4.

Le colonne *categoryList* e *categoryGroupList* sono simili nel concetto e nella gestione. In questo caso si possono utilizzare gli one hot vector in una forma leggermente modificata. Poiché ad ogni organizzazione corrispondono in generale più di una *categoryList* e più di una *categoryGroupList*, il vettore che assoceremo a ciascuna organizzazione avrà settati ad 1 i valori corrispondenti alle *categoryList* e *categoryGroupList* presenti nella sua descrizione, a zero gli altri. Vi sono tuttavia 603 possibili valori di *category* e 46 possibili valori di *categoryGroupList.* La loro gestione pertanto è complessa poiché va ad aumentare di molto la dimensione della matrice. Vedremo più avanti una proposta di metodo per utilizzare questa informazione in modo più compatto.

Le colonne *countryCode, satateCode, region* e *city* portano al loro interno un'informazione di natura geografica. L'utilizzo dello one hot encoding è possibile ma sconsigliato in questo caso per tre motivi. Il primo è ovviamente l'aumento vertiginoso

della dimensione della matrice. Il secondo è la distribuzione dei valori *"NA":* possiamo infatti notare, osservando nuovamente la Figura 5, che i campi *countryCode* e *city* sono sempre entrambi presenti o entrambi assenti, il che corrisponde al concetto intuitivo che ogni città del mondo si trova in una nazione. Tuttavia, non tutte le nazioni sono organizzate in stati e/o regioni, ad esempio per nazioni come l'Italia il valore *stateCode* sarà sempre *"NA"*. Fortunatamente, il terzo motivo per cui non utilizzeremo gli *one hot vector* è che disponiamo di una notazione molto più conveniente: quella in coordinate geografiche bidimensionali, nel formato latitudine e longitudine.

4.1.1 Ottenere le coordinate geografiche tramite i servizi Google

Nel nostro database contiamo più di 7000 differenti valori da convertire, pertanto l'uso delle variabili indicatrici nel formato *one hot vector* non è l'ideale. Per utilizzare le informazioni geografiche vogliamo convertire una stringa in una coppia di numeri che rappresentino latitudine e longitudine. La stringa sarà composta dalla concatenazione di 4 valori: *country_code*, *state_code*, *region* e *city.*

Chiaramente, questa operazione deve essere automatizzata e per fare ciò possiamo appoggiarci ai servizi Google. Il linguaggio Python dispone di una libreria apposita per Google Maps, la quale contiene dei metodi appositi per convertire una stringa indirizzo nella corrispondente coppia latitudine-longitudine.

Il livello gratuito di Google Cloud Platform (Google inc., s.d.) consente di utilizzare gratuitamente delle risorse. È necessario possedere una *API key* associata ad un account Google ed abilitare la fatturazione su ogni progetto. Un'*API key* (chiave di interfaccia per la programmazione delle applicazioni) è un codice utilizzato dal fornitore di servizi informatici per identificare il programma richiedente, il suo sviluppatore e/o un utente. Le API key vengono utilizzate per tenere traccia e controllare la modalità di utilizzo dell'API, ad esempio, per evitarne un uso dannoso o scorretto. Agisce sia come identificatore univoco, sia come token segreto per l'autenticazione e dispone di una serie di diritti di accesso ad esso associati.

La prova gratuita, della durata di un anno, fornisce un credito iniziale di 300 dollari. Il prezzo per ogni singola richiesta di conversione è 0.005 USD (5.00 USD per 1000) ma le prime 40000 richieste sono gratuite. Pertanto, per il nostro task è stato possibile convertire gratuitamente gli indirizzi. Ovviamente, questa operazione è stata eseguita solo una volta. Il dataset è stato ordinato per i valori di *city* in ordine alfabetico decrescente. Una volta ottenute le coordinate di una città, sono state aggiornate tutte le entry con lo stesso valore, per minimizzare il numero di richieste.

Il risultato di ogni traduzione è stato salvato nel file *companies_MKT_anonym.csv* aggiungendovi due nuove colonne: *organization_latitude* ed *organization_longitude.*

4.1.2 Selezione della category group list con rete neurale

Associare un'azienda a delle categorie corrisponde a descrivere sinteticamente di cosa si occupa. Come già detto, ogni categoria è associata a dei raggruppamenti, le *category_grouplist*. Ogni azienda del nostro database appartiene al raggruppamento sales

and marketing. Tuttavia, è facile intuire che le aziende hanno molto spesso delle risorse allocate per il marketing, quali personale e uffici, tali per cui rientrano in questa macrocategoria anche quando il core business è di tutt'altro tipo.

Ci chiediamo se, manipolando le informazioni in nostro possesso, sia possibile individuare qual è la macrocategoria in cui ogni azienda di rispecchia di più.

Il file *category_groups.csv* associa ad ogni categoria un insieme di macrocategorie. Osservando attentamente il contenuto dei campi *organization_categoryList* e *organization_categoryGroupList* del file *companies,* possiamo notare che *categoryGroupList* altro non è che l'insieme formato da tutti i raggruppamenti individuati dalle *category*. Essendo un insieme, i valori compaiono solo una volta, ma in generale alcuni valori potrebbero comparire più volte.

organization_categoryList	organization_categoryGroupList
business intelligence|graphic design|marketing|product design|ux design|web design	data and analytics|design|software|sales and marketing

Tabella 8: esempi di categoryList e categoryGroupList estratto da companies

category_name	category_groupList
business intelligence	data and analytics
graphic design	design
marketing	sales and marketing
product design	design
ux design	design
web design	design|software

Tabella 9: Group list associate alle category

Pertanto, se tra tutti i valori ne individuassimo uno la cui cardinalità è maggiore di tutti gli altri, potremmo affermare che quella è la *category grouplist* più significativa per l'azienda. Si vedano gli esempi in Tabella 8 e in Tabella 9: *design* compare 4 volte, mentre le altre grouplist soltanto una. Intuiamo quindi a qual è l'occupazione principale di quest'azienda.

Conteggiare la ripetizione delle grouplist e scegliere la più frequente potrebbe non essere sempre possibile. Potremmo avere dei casi di parità: se due grouplist compaiono con la stessa cardinalità, quale delle due dovremmo scegliere? O peggio, se tutte comparissero con cardinalità pari a 1? Abbiamo un'alternativa al fare una scelta puramente casuale?

Poiché per un sottoinsieme significativamente grande di aziende possiamo fare una scelta deterministica, queste possono costituire un training set per l'allenamento di un modello. Possiamo valutare il modello e, se i risultati saranno promettenti, affidarci a questo per la selezione di una grouplist in caso di incertezza. La speranza è che il modello usi la grande mole di dati a disposizione per imparare qualcosa sulla struttura di questo problema che gli sia utile per scegliere opportunamente una grouplist.

Come possiamo valutare le prestazioni di questo modello? Decidiamo che dovrà ricevere come input il vettore di variabili dummy associato a *category_list* e che dovrà restituire un altro vettore dummy in cui l'unico valore 1 è associato alla grouplist scelta. Sappiamo quali sono le possibili grouplist, ma non sappiamo quale scegliere. Allora potremmo associare alla scelta di una qualunque grouplist tra quelle presenti un successo e alla scelta di una grouplist non presente un fallimento. Considerano la struttura del nostro problema, la rete neurale si presenta come il mezzo più accattivante per la costruzione del modello

4.2 Cleaning dei dati

Per la sua natura *crowd source,* il dataset di Crunchbase è molto sparso. Ne abbiamo avuto conferma osservando la Figura 5.

In presenza di valori mancanti in un dataset, si presentano diverse opzioni. È possibile fare delle assunzioni sui dati ed inserire dei valori fittizi. Un comune approccio consiste nel calcolare ed utilizzare il valore medio dei vari campi.

Nel nostro caso, osservando i dati su un foglio di calcolo, notiamo che al valore di funding rounds pari a zero, corrisponde un valore *"NA"* in funding total USD. Correggiamo manualmente le entry in questi casi, poiché chiaramente a "NA" corrisponde il valore 0.

Non ci sembra lecito nella nostra analisi utilizzare dati incompleti. Procediamo con il cleaning, eliminando i valori *"NA".* Pandas dispone per questo del metodo *.dropna().* Andiamo quindi ad analizzare il dataset risultante. Se il numero di campioni si rivelerà sufficientemente ampio, potremo decidere di non fare ulteriori assunzioni.

```
1  companies=companies.dropna()
2  companies.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 34317 entries, 0 to 46402
Data columns (total 9 columns):
Status              34317 non-null object
Foundation year     34317 non-null float64
Cat List            34317 non-null object
Cat GruopList       34317 non-null object
Funding rounds      34317 non-null int32
Funding total $     34317 non-null float64
Employees           34317 non-null object
Latitude            34317 non-null float64
Longitude           34317 non-null float64
dtypes: float64(4), int32(1), object(4)
memory usage: 2.5+ MB
```

Figura 6: dataset risultante dopo l'eliminazione dei valori NA

Osserviamo che il dataset si è quasi dimezzato. Tuttavia, il numero di campioni è molto consistente, pertanto ci consideriamo soddisfatti. Andiamo ad analizzare i campioni rimasti più nel dettaglio.

Poiché il nostro obiettivo è classificare le startup per *exit* o *closed,* iniziamo con il semplice conteggio dei campioni disponibili. Conteggiamo anche quante organizzazioni hanno ricevuto almeno un funding round (FR). Riportiamo i risultati in Tabella 10.

	Totale	**Percent.**	**No FR**	**Percent.**	**FR > 0**	**Percent.**
Operating	28782	83,87%	23408	81,32%	5374	18,68%
Exit	3975	11,58%	2669	67,14%	1306	32,86%
Closed	1560	4,55%	1029	65,96%	531	34,04%

Tabella 10: Distribuzione per status e investimenti ricevuti

La grandissima maggioranza delle compagnie si trova in stato *operating*. Di queste, la grande maggioranza non ha mai ricevuto investimenti. Può essere il caso di startup giovani, ma più probabilmente questo rispecchia la realtà delle aziende di piccole-medie dimensioni. Queste spesso individuano la loro nicchia di mercato e la conservano per molti anni. In questo caso, possono non avere particolare interesse o non avere nemmeno la possibilità di quotarsi in borsa, anche se complessivamente possiamo argomentare che abbiano avuto una forma alternativa di successo. Purtroppo, questa forma di successo è più difficile da definire rigorosamente.

Osserviamo le organizzazioni per le quali il label viene fornito, ovvero il nostro training set: consiste in totale di 5535 organizzazioni. Questi è inoltre profondamente sbilanciato: solo il 28% delle organizzazioni sono *closed*. I dataset sbilanciati sono problematici per gli algoritmi di machine learning. Un algoritmo potrebbe riconoscere questo sbilanciamento e optare per assegnare il label “exit” a qualsiasi input gli venga fornito. Se il nostro training set rispecchia correttamente la distribuzione di probabilità che lo ha generato, questo algoritmo indovinerà approssimativamente nel 72% dei casi senza aver imparato nulla sui dati. Per risolvere il problema dello sbilanciamento, possiamo semplicemente eseguire un *subsampling:* selezioneremo un sottoinsieme casuale di organizzazioni in stato *exit* di cardinalità pari all’insieme *closed*. In questo modo bilanceremo il dataset al prezzo della perdita di campioni, ma renderemo i risultati dell’analisi più semplici da interpretare. Un classificatore che fallisce nell’apprendimento avrà in quel caso errore prossimo al 50%.

È lecito domandarsi se questa distribuzione così sbilanciata rispecchi la realtà. La nostra conoscenza del mondo reale ci suggerisce che probabilmente le startup che chiudono siano molte di più rispetto a quelle che arrivino a fare una IPO o vengano acquisite. Possiamo ipotizzare che, data la natura aperta e finalizzata a scopo autopromozionale di Crunchbase, le startup in difficoltà, o che hanno già chiuso, non abbiano alcun interesse a condividere i propri dati di andamento negativo. D’altra parte, questo ragionamento decade nel momento in cui ricevono un investimento, a quel punto le informazioni sono pubblicamente accessibili e gli stessi moderatori di Crunchbase possono inserirle ed aggiornare il database.

4.3 Feature Engineering

Le colonne candidate di *companies* sono dunque le seguenti: *status, category List, category grouplist foundedOnYear, fundingRounds, fundingTotalUsd, employeeCount, latitude, longitude.* Vediamo in questa sezione come utilizzare gli altri file a nostra disposizione per arricchire il dataset.

Le variabili aggiuntive da noi introdotte manipolano i dati grezzi a nostra disposizione per produrre informazione strutturata. L'obiettivo di questo arricchimento è fornire agli algoritmi nuove variabili portatrici di informazione significativa utili al task di classificazione.

4.3.1 Burn rate

Il *burn rate* (Kenton, 2019) di un'organizzazione è il termine tipicamente utilizzato per indicare il tasso a cui una nuova società spende il suo capitale di rischio per finanziare le spese generali prima di generare un flusso di cassa positivo. Si tratta di una misura del flusso di cassa negativo. Il *burn rate* è solitamente indicato in termini di liquidità spesa mensilmente. Un'azienda con *burn rate* di 1 milione di dollari, spende questa cifra ogni mese. Utilizzando i campi *organization_foundedOn, organization_totalFundingUsd* e *organization_lastFundingOn* della tabella companies ed implementando un'opportuna funzione che calcoli, fornite due date in formato gg/mm/aaaa, il numero di mesi trascorsi, possiamo calcolare in maniera approssimata il *burn rate* medio delle aziende nel nostro database. Non abbiamo informazioni circa il flusso di cassa positivo delle aziende, perciò il valore che calcoliamo assume implicitamente che l'azienda spenda tutto il suo capitale.

4.3.2 Variazione di burn rate

Poiché disponiamo dello storico degli investimenti, nel file *fundingRounds_MKT_anonym.csv,* possiamo calcolare il tasso medio di variazione di *burn rate.* Data un'azienda, in ordine cronologico per ogni investimento, calcoliamo il *burn rate* e lo dividiamo per il tempo trascorso. Facciamo infine una media di questi valori.

Questa feature ci dice se la spesa di denaro mensile di un'impresa nel tempo è stata soggetta ad un'accelerazione o ad un rallentamento. Come interpretiamo i valori di questo dato? Possiamo ipotizzare che per un'impresa giovane, una significativa accelerazione di burn rate indichi una crescita: corrisponde all'assunzione di personale e all'acquisto di asset. Per un'impresa attiva da più tempo invece, una crescita modesta o nulla potrebbe indicare che il flusso di cassa positivo è ormai sufficiente per autosostenerla (ricordiamo che su questo flusso non abbiamo informazioni e stiamo supponendo che le aziende consumino tutto il denaro ricevuto in investimenti) e che necessita di investimenti più modesti da parte di terzi.

4.3.3 Investimenti ripetuti (fiducia degli investitori)

Infine, poiché disponiamo dell'identità (anche se anonima) degli investitori, possiamo contare il numero di volte in cui lo stesso investitore ha investito in una certa compagnia. Questa misura di *fiducia* può essere interpretata in due modi. In senso positivo, l'investitore è convinto del successo della compagnia, per questo ripete gli investimenti su di essa. In senso negativo, un investitore potrebbe finanziare ulteriormente un'azienda per evitarne il fallimento. Aggiungeremo quindi una colonna al dataset il cui valore sarà il conteggio complessivo dei re-investimenti da parte dei vari investitori. Intuitivamente, se un alto valore di fiducia è accompagnato da un valore positivo di variazione di *burn rate*, significa che l'azienda sta crescendo e viene attivamente sostenuta nella sua crescita, mentre un valore di fiducia basso, ma comunque superiore a zero, accompagnato da accelerazione negativa, può essere un segnale non incoraggiante: l'investitore potrebbe stare cercando di evitare il fallimento dell'impresa per salvare il suo primo investimento.

4.4 Statistica descrittiva

Ricevere un finanziamento è una milestone molto importante per un'azienda ed è sempre più difficile riceverne altri dopo il primo. I grafici in Figura 7 e Figura 8 mostrano un chiaro andamento esponenziale. Notiamo meglio, dal secondo grafico, che tra le aziende *closed* con almeno un finanziamento, più della metà hanno chiuso dopo il primo investimento. L'unica colonna per cui il numero di *closed* si avvicina al numero di *exit* è proprio quella corrispondente alle aziende che hanno ricevuto un solo investimento. Per tutte le altre, le aziende di successo sono nettamente in maggioranza.

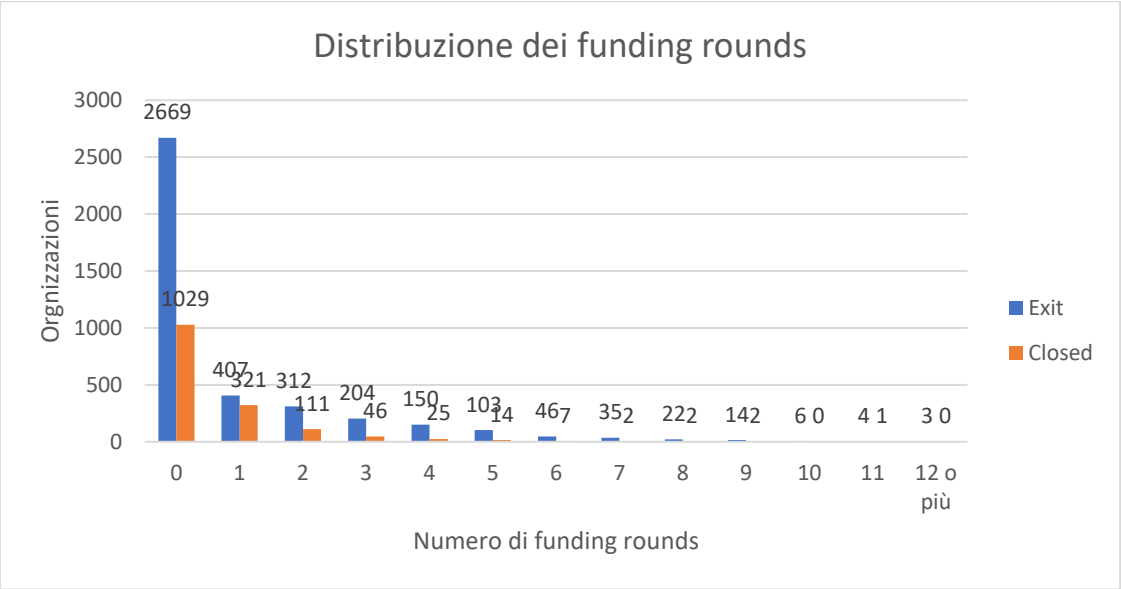


Figura 7: Funding round per impresa

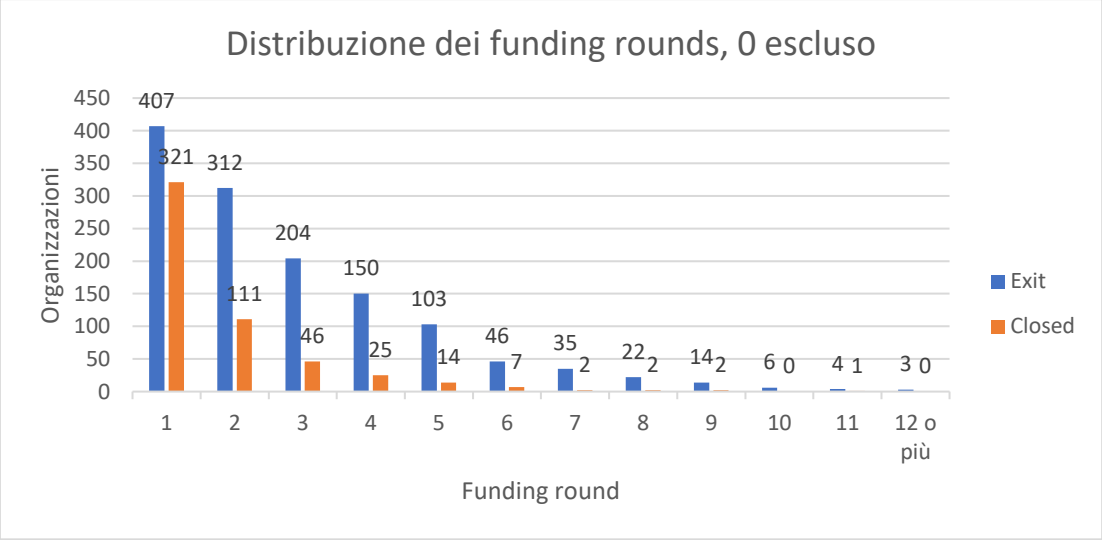


Figura 8: Funding round per impresa

Utilizziamo le informazioni geografiche per plottare in un grafico le organizzazioni in *exit* (in verde) e le *closed* (in rosso). La Figura 10 ci dà informazioni interessanti. Dal grafico riusciamo chiaramente a identificare Europa e nord America. Più a grandi linee riconosciamo la penisola indiana, parte di Brasile, Argentina, Nuova Zelanda, Giappone, la costa est di Cina ed Australia. Alcuni punti particolarmente densi corrispondono ad Israele, Taiwan e Hong Kong. Vediamo che alcune zone sono sbilanciate verso un colore in particolare. Questo corrisponde all'intuizione dell'esistenza di zone "privilegiate" in cui il clima imprenditoriale favorisce maggiormente la nascita e lo sviluppo di startup. Ma questo ci da anche un'idea della diffusione internazionale dell'utilizzo di Crunchbase.

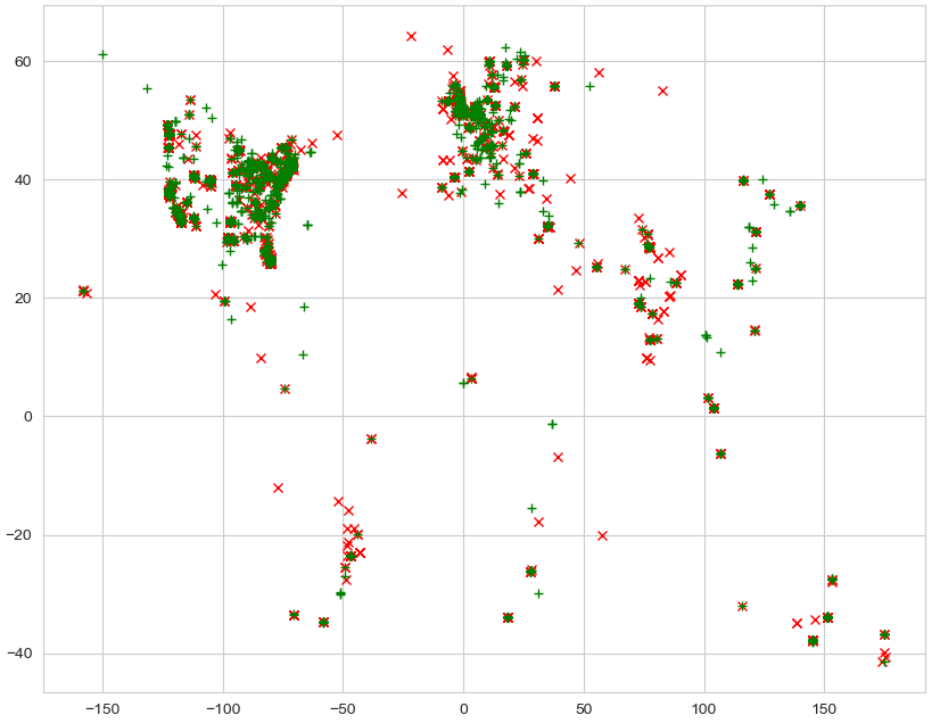


Figura 10: distribuzione delle startup nel mondo. In verde "ipo" e "acquired", in rosso "closed"

5 Metodologia ed ipotesi di ricerca

Il presente lavoro si inserisce all'interno di un filone di ricerca che negli ultimi anni ha riscosso crescente interesse da parte della comunità scientifica.

Citiamo due interessanti articoli correlati a questa tesi. In entrambi gli autori hanno utilizzato tecniche di machine learning per analizzare i dati forniti da Crunchbase, con scopi e risultati differenti.

Il primo, *Predicting investor funding behavior using crunchbase social network features* (LIANG & YUAN, 2016) tenta di predire il comportamento degli investitori nei confronti delle organizzazioni costruendo un *social network graph* (Brilliant.org, s.d.), ovvero una struttura atta a rappresentare la rete di conoscenze e contatti che collega investitori ed aziende. In questo articolo quindi, il dataset originale è stato oggetto di una notevole e sofisticata elaborazione, il cui risultato, il social network graph, consiste di una nuova entità, con una diversa struttura e significato, sulla quale è stato possibile utilizzare con successo gli algoritmi di learning e regressione.

La seconda, *Text based classification of companies in CrunchBase* (BATISTA & CARVALHO, 2015), utilizza il metodo detto *fuzzy fingerprints* (STEIN, 2005) per classificare le organizzazioni sulla base della descrizione testuale inserita in Crunchbase, distinguendo più di 40 possibili etichette di classificazione per le organizzazioni. In questo caso si è utilizzato il dato originale in combinazione con algoritmi di *information retrieval* e machine learning.

5.1 Domande di ricerca

Il nostro obiettivo è utilizzare algoritmi di machine learning per suggerire agli investitori quali siano le imprese più promettenti su cui investire.

Cercheremo di etichettare delle imprese in stato *"operating"*, ovvero attualmente in attività, con uno tra i due possibili label *"exit"* e *"closed"*, i quali rappresentano, in prima approssimazione, l'indicatore di successo o fallimento della startup.

L'obiettivo della nostra analisi sarà quindi quello di individuare quale, tra vari modelli esistenti, sia il più adatto a questo task. Cercheremo di capire quali informazioni danno il contributo più incisivo alla scelta. Osserveremo come combinare e manipolare i dati per produrre nuove informazioni utili e misureremo l'impatto di questi nuovi dati sui classificatori.

5.2 Significatività della metodologia

La metodologia utilizzata, oltre ad essere un approccio innovativo di recente formulazione ed a produrre come risultato un modello predittivo, presenta anche altri vantaggi rispetto ai metodi di analisi statistica tradizionale.

Il dataset a nostra disposizione consiste di entità molto diverse tra loro. Esse rappresentano startup, compagnie, persone, movimenti di denaro ed organizzazioni finanziarie. Abbiamo informazioni di carattere temporale, spaziale e quantitativo. L'eterogeneità e ricchezza di queste informazioni costituisce la principale risorsa e sfida per l'analisi. I dati dovranno essere normalizzati e trasformati da stringhe di testo in vettori di numeri. Questi vettori rappresenteranno punti in uno spazio n-dimensionale. A questo punto i problemi concettuali esposti finora si trasformano in problemi geometrici. È ora possibile sviluppare ipotesi, esperimenti e misure di valutazione precisi, rigorosi e riproducibili. Valuteremo prima le potenzialità del dataset originale, per poi arricchirlo con funzioni che, a partire dalla nostra intuizione, andranno a produrre nuove informazioni utili a classificare le startup.

Tutti i task di machine learning condividono la stessa ipotesi di partenza: è possibile imparare qualcosa a partire dai dati a disposizione. In alcuni casi questo significa acquisire nuove conoscenze sui dati stessi, non individuabili a prima vista e di interesse significativo. In altri, significa poter utilizzare l'informazione già in nostro possesso come fonte di esperienza finalizzata allo svolgimento di un certo compito. Pertanto, nel nostro caso, sviluppare delle ipotesi si traduce in definire uno o più compiti per i quali allenare dei modelli con i dati a disposizione e valutarne poi le performance.

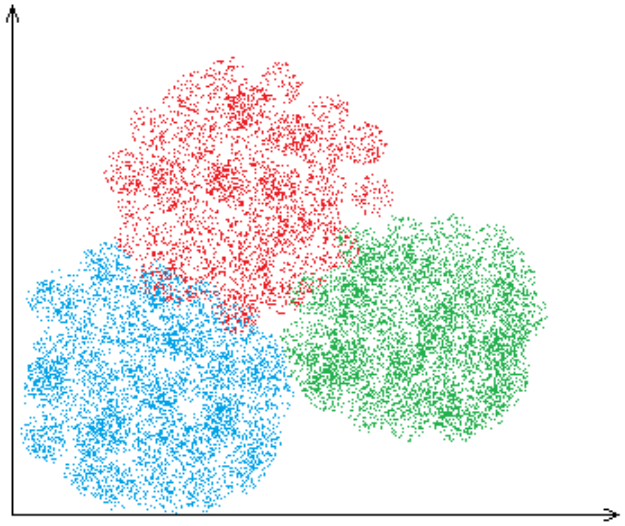
L'analisi qualitativa dei dati della sezione precedente ha guidato la scelta dei modelli e algoritmi da utilizzare.

5.3 Algoritmi utilizzati

Il dataset consiste di una matrice di *m* righe ed *n* colonne, la quale rappresenta un insieme di *m* punti distribuiti in uno spazio di dimensione *n*. Possiamo visualizzarla come una nuvola di punti in uno spazio $\mathbb{R}^n$. Nel modello di learning, l'ipotesi a monte è che i punti non siano distribuiti casualmente ma secondo una distribuzione di probabilità sconosciuta, $\mathcal{D}$. Avendo a disposizione una grande quantità di campioni, il nostro obiettivo è utilizzarli per "imparare" $\mathcal{D}$, o almeno approssimarla con sufficiente grado di confidenza.

5.3.1 K-means clustering

Figura 9: Esempio con 3 cluster di punti



Parliamo di *unsupervised learning* quando ai punti non è associato un *label:* un'etichetta che li identifichi come appartenenti ad uno specifico insieme. L'approccio di studio più comune in questi casi è detto *clustering.*

Nei problemi di clustering si cerca di individuare degli agglomerati (cluster), ovvero dei raggruppamenti significativi e distinguibili di punti in certe zone dello spazio. In Figura 9 vediamo un esempio con tre cluster ben definiti in R^2. Intuiamo che i punti nei vari cluster, qui distinti per colore, sono in qualche modo più simili ai punti nello stesso cluster che a quelli appartenenti ad altri cluster.

In machine learning, quando si esegue un clustering l'ipotesi di fondo è sempre la stessa: i punti che condividono lo stesso cluster probabilmente appartengono alla stessa classe. Possono esserci più cluster che formano una singola classe. La parola "probabilmente" appare nella definizione poiché non è nota a priori una metodologia che discrimini se l'ipotesi è valida o meno. Inoltre, anche il significato di classe non è univoco. In generale, si assume che si possano associare label diversi ai punti appartenenti a cluster diversi. Infine, attribuire un label numerico ad un cluster non è sufficiente ad attribuirgli un significato. Se individueremo dei cluster ben definiti nel nostro dataset, dovremo capire le caratteristiche che contraddistinguono le aziende appartenenti ai vari cluster.

Per risolvere un problema di clustering è necessario individuare il numero ottimale di cluster e i relativi centroidi. Molte tipologie di cluster sono possibili e possono dare risultati molto differenti l'una dall'altra. Purtroppo, individuare il cluster ottimo per uno specifico dataset, ovvero quello che ne catturi meglio le caratteristiche, è un problema appartenete alla classe NP (CORMEN & al., 2009, p. 1064-1067) e quindi non risolvibile efficientemente con i calcolatori disponibili oggi. Pertanto, si ricorre ad

algoritmi euristici di approssimazione i quali tentano di produrre delle soluzioni il più possibile vicine alla soluzione ottima.

L'algoritmo che noi utilizzeremo è detto *k-means clustering,* algoritmo originariamente proposto da Stuart Lloyd nel 1957. Questo algoritmo riceve in input il valore *k* di cluster da utilizzare ed inizializza casualmente *k* centroidi. Quindi assegna ogni punto del dataset al cluster più vicino ed aggiorna le coordinate di ogni centroide come media dei punti appartenenti al cluster da esso individuato.

Ci rendiamo conto delle limitazioni di questo algoritmo: il numero di cluster e la scelta dei centroidi, essendo casuali, potrebbero non essere adeguati al dataset. Inoltre, non è univoca la scelta della condizione di terminazione dell'algoritmo, anche se, in generale, si sceglie di interromperlo dopo un certo numero di iterazioni oppure quando lo spostamento dei centroidi tra un'iterazione e l'altra diviene inferiore ad una certa soglia. Il vantaggio di *k-means* è la velocità di esecuzione, grazie alla quale è possibile ovviare agli altri problemi ripetendo l'esecuzione dell'algoritmo. Dunque, per ogni scelta di *k,* l'esecuzione verrà ripetuta per un numero fissato di volte, verrà valutato il cluster risultante e si sceglierà quello considerato migliore.

Altro vantaggio notevole è che per memorizzare un cluster è sufficiente memorizzare la posizione dei centroidi.

Per scegliere il valore di *k* ottimale per il clustering. Utilizzeremo due metodi, la curva di Elbow ed il coefficiente di silhouette.

Elbow curve

La curva di Elbow (KODINARIYA & MAKWANA, 2013) è un metodo euristico di interpretazione e validazione della consistenza e coerenza tra cluster, progettato per aiutare ad individuare graficamente il numero appropriato di cluster in un dataset (Figura 10). Poiché richiede un'interpretazione intuitiva, spesso è ambiguo e poco affidabile rispetto al coefficiente silhouette. Questo metodo prende in considerazione la percentuale di varianza, intesa come il rapporto tra la varianza tra gruppi e la varianza totale, in funzione del numero di cluster: la scelta del numero di cluster dovrebbe essere tale per cui l'aggiunta di un altro cluster non dia una migliore modellazione dei dati. Più precisamente, se si traccia la percentuale di varianza indotta dai cluster rispetto al loro numero, i primi aggiungeranno molte informazioni (quindi si osserverà una significativa variazione percentuale), ma ad un certo punto il guadagno marginale diminuirà, dando origine nel grafico ad una variazione angolare più marcata. Il numero di cluster viene scelto guardando per quale *k* si ha questo "ginocchio" (elbow).

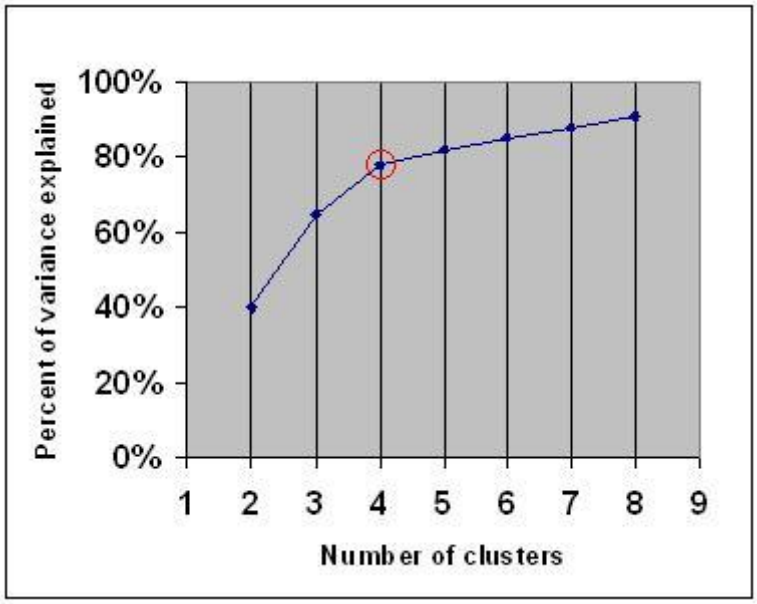


Figura 10: Curva di Elbow ideale. In rosso il punto di ginocchio

Silhouette

Il coefficiente di *silhouette* (Rousseeuw, 1987) misura indicativamente quanto un oggetto è vicino ad oggetti nel suo stesso cluster rispetto ad oggetti appartenenti ad altri cluster. Il coefficiente di silhouette è un valore compreso tra -1 e 1. Un punto con valore di silhouette prossimo ad 1 è circondato da punti appartenenti allo stesso cluster. Un valore prossimo a 0 indica che il punto si trova al confine tra un cluster e l'altro mentre valori negativi stanno a indicare che sarebbe più appropriato assegnare il punto ad un altro cluster. La media di tutti i valori di silhouette indica quanto i punti siano strettamente raggruppati tra loro. Un alto valore di silhouette per qualche *k* è indice della presenza di una struttura coerente.

5.3.2 Support Vector Machine (SVM)

Nei problemi di classificazione, si cercano di individuare uno o più iperpiani che dividano lo spazio in regioni alle quali associamo dei label, etichette. In due dimensioni, un iperpiano corrisponde ad una funzione, come una retta od un polinomio, che divide lo spazio in due regioni, associate ciascuna ad un label. Il problema decisionale si traduce nel problema di assegnare ad un punto l'etichetta corretta, in base a dove si trova nello spazio. In Figura 11 vediamo un esempio. Un algoritmo ha elaborato 3 ipotesi per classificare i dati: H1, H2 ed H3. Come sceglierà quale restituire come output? Possiamo misurare l'errore commesso dalle ipotesi, contando quanti campioni vengono etichettati con il label sbagliato. Vediamo che H1 classifica erroneamente cinque dei punti etichettati in nero, mentre H2 ed H3 classificano correttamente tutti i punti. Come scegliamo una delle due? Il nostro intuito ci dice che H3 è la scelta migliore, perché H2 è in qualche modo troppo vicina ai dati, mentre H3 ha più *margine*. L'algoritmo SVM, *support vector machine,* formalizza questa intuizione.

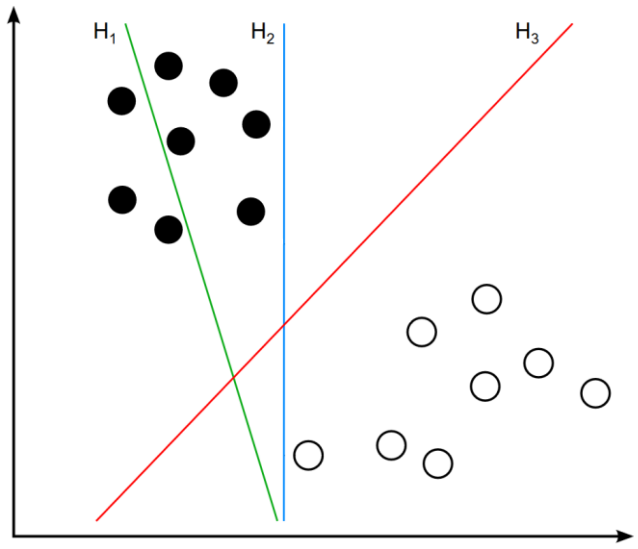


Figura 11: Esempi di iperpiani separatori, l'algoritmo cerca di selezionare quello che meglio classifica i dati

L'algoritmo che noi utilizzeremo è noto col nome di SVM: *Support Vector Machine* (SHALEV-SHWARTZ & BEN-DAVID, 2014, p. 202-214). È particolarmente adatto a risolvere problemi in cui l'input ha una dimensione *m* molto grande. È un problema di ottimizzazione convesso, quindi può essere risolto efficientemente in tempo polinomiale. Questo algoritmo non si limita a scegliere un iperpiano che classifichi correttamente il maggior numero possibile di punti. Introduce infatti il concetto di margine, inteso come distanza tra i campioni e l'iperpiano separatore, e cerca di massimizzarlo. I punti più vicini all'iperpiano sono detti vettori di supporto *(support vector)* e danno il nome all'algoritmo.

5.3.3 Random Forest

Random Forest (RF) è un insieme di alberi decisionali (SHALEV-SHWARTZ & BEN-DAVID, 2014, p. 250 - 255). Un albero decisionale altro non è che un diagramma di flusso del processo decisionale per la scelta della label, basato sulle feature del dataset.

Il processo è rappresentato con un albero logico rovesciato. Ogni nodo è una funzione condizionale che verifica una condizione su una particolare feature ed ha due o più diramazioni verso il basso. Il processo comincia sempre dal nodo radice e procede verso il basso. L'output del modello si trova nei nodi foglia terminali.

Rispetto ad altri algoritmi di machine learning, il modello prodotto dall'albero decisionale è comprensibile dagli esseri umani. Pertanto, possiamo verificare come la macchina assegna il label. Questo rende possibile osservare alcuni criteri decisionali adatti alla logica delle macchine ma scarsamente comprensibili dall'uomo.

La rappresentazione ad albero decisionale è tuttavia poco adatta per i problemi complessi. Il numero di rami in uscita da un nodo è pari al numero di possibili valori che il nodo può assumere. Se già questo è un problema nel campo dei numeri interi, in presenza di una o più feature a valori reali lo scenario può diventare critico. Lo spazio delle ipotesi potrebbe in questo caso crescere esponenzialmente e, con esso, la complessità spaziale dell'algoritmo.
Nel nostro caso abbiamo variabili miste: booleane, multilabel e reali. Cosa possiamo fare?

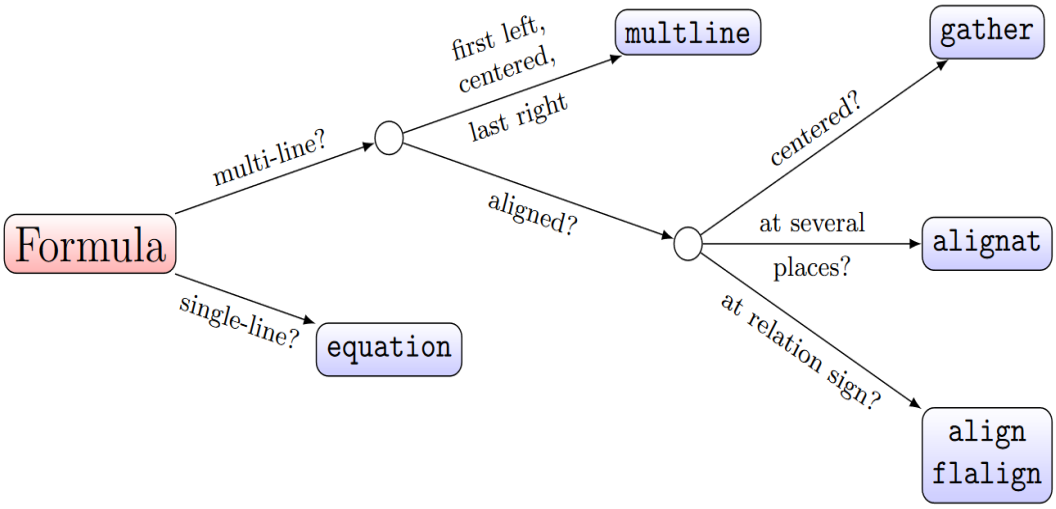


Figura 12: Esempio di decision tree

Quando parliamo di Ensemble learning, in generale, ci riferiamo ad un modello che fa previsioni basate su diversi modelli. Combinando i singoli modelli, il modello ensemble tende ad essere più flessibile e ad avere meno varianza.

Due dei metodi di ensemble learning più popolari sono il *bagging* ed il *boosting* (SHALEV-SHWARTZ & BEN-DAVID, 2014, p. 130 - 141).

Con il termine *bagging* indichiamo l'allenamento parallelo di singoli modelli, allenati con un sottoinsieme casuale dei dati.

Il *boosting* consiste invece nell'allenamento di un gruppo di singoli modelli in modo sequenziale. Ogni modello impara dagli errori del modello precedente.

Random Forest utilizza il *bagging* allenando un insieme di alberi decisionali, ciascuno su un sottoinsieme diverso di colonne del dataset, campionato casualmente. RF tenta di prevenire il sovraffollamento di rami dell'albero creando sottoinsiemi casuali di feature e costruendo alberi più piccoli (e quindi poco profondi).

In presenza di un nuovo campione da classificare, RF lo valuta con tutti i suoi alberi e sceglie come output il label comparso più volte.

Un importante vantaggio di RF è che fornisce stime di quali variabili sono più importanti per la classificazione. Lo svantaggio principale, rispetto ad un semplice albero decisionale, è la perdita di interpretabilità: è difficile comprendere la relazione tra una feature e l'insieme di regole create.

AdaBoost (*adaptive boosting*) funziona particolarmente bene con l'albero decisionale. L'obiettivo del modello di Boosting è imparare dagli errori precedenti. Per fare questo, AdaBoost assegna dei pesi ad ogni albero decisionale nella random forest. I pesi sono basati sul training error di ogni singolo albero. Tanto minore è il training error dell'albero, tanto maggiore sarà il suo peso, e con esso il suo potere decisionale. Il label finale è ottenuto sommando la previsione di ciascun albero moltiplicata per il suo peso. L'albero con peso maggiore avrà più potere di influenzare la decisione finale.

5.3.4 Neural Network

Una rete neurale (SHALEV-SHWARTZ & BEN-DAVID, 2014, p. 268 - 281) è un tipo particolarmente avanzato di classificatore. Si tratta un grafo aciclico diretto, i cui nodi, chiamati anche neuroni, sono organizzati in *layer*. Il primo e l'ultimo sono detti rispettivamente *input layer* ed *output layer*. Tra questi vi è un certo numero di *hidden layer*, o layer nascosti, si veda Figura 13. L'*input layer* ha un numero di nodi pari alla dimensione dei vettori usati come input per allenare la rete, mentre la dimensione dell'*output layer* dipenderà da che tipo di classificatore vogliamo implementare. Gli *hidden layer* sono molto importanti. Il numero di layer e il numero di nodi per ogni layer costituiscono l'architettura della rete. L'architettura deve essere scelta in fase di progettazione. Essa determinerà l'espressività della rete neurale.

Una rete neurale è in grado dividere lo spazio non soltanto in iperpiani ma anche in strutture più complesse. Possiamo visualizzare questo concetto con degli esempi nello spazio $\mathbb{R}^2$. Una rete neurale con un solo *hidden layer* è in grado di dividere un piano in semispazi (Figura 14 a), in maniera simile ad algoritmi quali SVM. Una rete con due *hidden layers* invece, può esprimere intersezioni di semispazi (Figura 14 b). Una con tre *hidden layers* può esprimere unioni di intersezioni di semispazi (Figura 14 c).

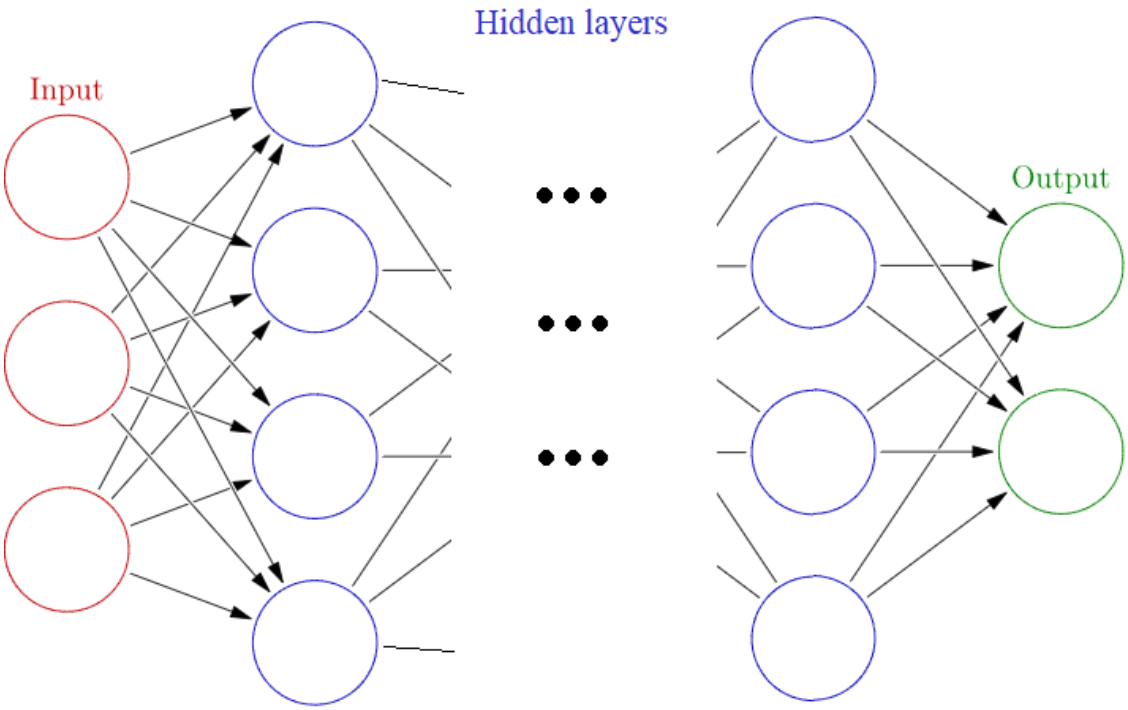


Figura 13: Schema di una rete neurale

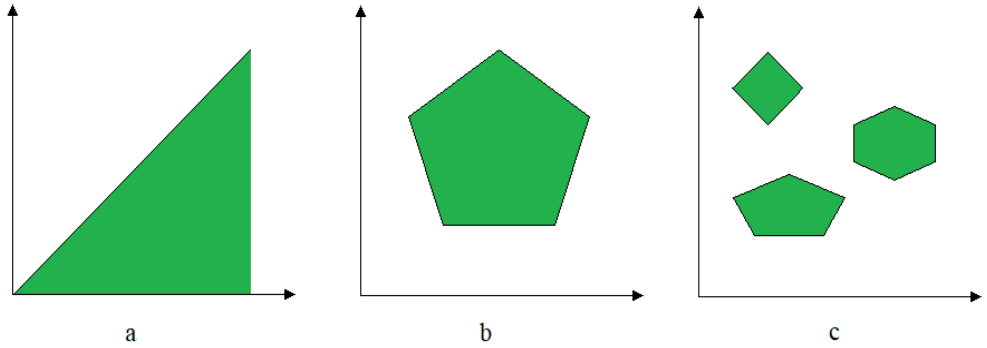


Figura 14: esempi di espressività di una rete neurale con uno (a), due (b) e tre (c) hidden layer.

Questo permette di classificare i punti in base a funzioni quali XOR, l'operatore booleano OR esclusivo, le quali non sono esprimibili da nessuna funzione lineare. Questa maggiore espressività ha tuttavia un costo in termini di complessità: la scelta della miglior architettura e l'allenamento della rete sono entrambi problemi NP.

Per aggirare questo problema, ricorriamo ad un algoritmo greedy, la *backword propagation*. Come *k-means* riceve in input il numero di cluster, questo algoritmo riceve in input l'architettura della rete. Questo costituisce il *bias induttivo* che i progettisti devono fornire. A questo punto l'algoritmo comincia inizializzando una rete "casuale" e, ricevendo in input dei campioni, la corregge in modo da classificarli correttamente. La casualità intrinseca nel metodo fa sì che anche in questo caso sia necessario procedere per tentativi, ripetendo l'esecuzione dell'algoritmo alla ricerca di un'architettura particolarmente efficacie.

In ultima, anche la dimensione del dataset necessario all'allenamento della rete è esponenziale nel numero di nodi. Le reti neurali hanno bisogno di quantità molto maggiori di dati rispetto ad altri algoritmi, il che si traduce anche in tempi di calcolo molto più lunghi. Le reti neurali sono *universal approximators,* sono cioè in grado di approssimare qualsiasi distribuzione di probabilità sconosciuta $\mathcal{D}$ purché abbiano accesso ad abbastanza dati e ad abbastanza risorse computazionali da provare e riprovare architetture diverse fino a trovare quella più opportuna. Per questo non violano il *no-free-lunch theorem:* non esiste un'architettura in grado di approssimare qualunque distribuzione, indipendentemente da quanto grande la scegliessimo. Anche se la individuassimo, sarebbe comunque possibile, per ogni problema, costruire un modello alternativo con prestazioni superiori per quel dato problema. Ecco perché non si utilizzano soltanto reti neurali, ma anche altre soluzioni ed altri modelli.

Assegneremo opportunamente delle etichette all'insieme dei punti a nostra disposizione, poi lo divideremo in due sottoinsiemi: training set e test set. Alleneremo i modelli con il training set, poi useremo i loro output per predire i label del test set. Confronteremo quindi le previsioni con i veri label, a noi noti. In questo modo possiamo stimare con precisione la percentuale di accuratezza dei nostri modelli, ovvero con che precisione abbiamo approssimato $\mathcal{D}$.

5.4 Ipotesi e risultati attesi

1. Eseguendo un cluster sul dataset originale, questo individuerà dei raggruppamenti significativi. Analizzando i raggruppamenti si potranno dedurre proprietà significative dei dati. Se questo non è possibile, ci aspettiamo dei valori di silhouette prossimi a zero o negativi.
2. Ipotizzando che sia possibile classificare correttamente le startup, i nostri algoritmi si approcceranno in modi differenti a questo compito. Verificheremo quale si dimostrerà il più adatto.
3. Confronteremo i risultati ottenuti con il dataset di partenza, con il dataset arricchito dalle feature burn rate (br), variazione di br ed investimenti ripetuti.

5.5 Metriche di valutazione

Per gli algoritmi di classificazione utilizzeremo le seguenti metriche di valutazione.

- *Accuracy*: indica la percentuale di accuratezza del modello. È una stima accurata del vero errore del modello.

- *Confusion Matrix*: si compone delle seguenti 4 entità:
  - True Positive Rate (**TPR**): percentuale di label "1" assegnate correttamente.
  - False Positive Rate (**FPR**): percentuale di label "1" assegnate erroneamente.
  - True Negative Rate (**TNR**): percentuale di label "0" assegnate correttamente.
  - False Positive Rate (**FNR**): percentuale di label "0" assegnate erroneamente.

  Nel nostro caso le label 0 – 1 corrispondono a "closed" ed "exit"

- $Precision = \frac{TPR}{TPR+FPR}$: è il rapporto le exit correttamente predette e il totale delle label predette exit.

  Misura quanto il modello è accurato nel prevedere una exit, esiste la corrispondente precision per le closed.

- $Recall = \frac{TPR}{TPR+FNR}$: è il rapporto tra le exit correttamente predette e il totale di exit.

  Recall esprime la frazione di exit che siamo riusciti ad individuare. Come per precision, esprimiamo il valore di recall anche per la label closed.

- $F1Score = 2 * \frac{Precision * Recall}{Precision + Recall}$: può essere interpretata come media armonica ponderata della precisione e del richiamo. Raggiunge il suo miglior valore a 1 e il peggiore a 0. Calcolata per entrambe le label.

- *AUC – Roc Curve*: AUC **(**Area Under The Curve**)** ROC **(**Receiver Operating Characteristics**)**

  È una misurazione delle prestazioni per problemi di classificazione. ROC è una curva di probabilità e AUC rappresenta il grado o la misura della separabilità: indica cioè quanto modello è in grado di distinguere tra le classi.

  La curva ROC viene tracciata come un grafico ascissa-ordinata, con il TPR sull'asse y ed l FPR sull'asse x.

6 Risultati sperimentali

6.1 Selezione della grouplist

Per produrre il training set per la rete neurale, costruiamo due matrici formate da vettori di variabili dummy: *traning set* e *label set.* Per ogni entry in company, eseguiremo il conteggio della grouplist più frequente. Le entry per le quali emerge un valore univoco costituiranno il training set.

Cominciamo costruendo gli one hot vector a partire dalle colonne di *companies.*

Analizzando il file *companies* ci accorgiamo che sono presenti in totale 703 categorie, mentre in *categories* ne sono presenti soltanto 680. Quindi per 23 categorie non abbiamo informazioni sulle grouplist. Possiamo gestire l'inconsistenza in due modi: eliminando queste categorie dal conteggio oppure introducendo una grouplist aggiuntiva, che chiameremo *unknown,* che raccolga questi elementi. Invece le grouplist sono coerenti e sono in totale 46.

Nel primo test, i valori inconsistenti saranno semplicemente ignorati nel conteggio. Il risultato dell'analisi è mostrato in Figura 16.

Notiamo diverse cose. Non tutti i 46 valori di grouplist possibili compaiono, ma soltanto 39. Riusciamo ad ottenere un valore univoco solo per circa metà del dataset. Inoltre, il valore *"sales and marketing"* compare 18383 volte, ovvero più della metà. È opportuno considerare il valore sales and marketing? Le aziende del dataset sono già state estratte filtrando per questa categoria, quindi in un serto senso l'informazione è ridondante. Visto che stiamo cercando di capire di cosa si occupano queste organizzazioni al di là dell'aspetto sales and marketing, proviamo a ripetere il conteggio escludendo questa grouplist. Otteniamo il secondo risultato riportato in figura.

Stavolta emerge una seconda categoria dominante: *advertising*. Un risultato accettabile, essendo la grouplist più concettualmente affine a *sales and marketing*. Possiamo dirci soddisfatti del dataset? Idealmente per la rete neurale vorremmo una distribuzione il più possibile omogenea dei label. Inoltre, alcune entry individuano come grouplist soltanto *sales and marketing*. Se questo valore viene eliminato non resta alcun modo per classificarle se non un assegnamento puramente casuale (ed in definitiva errato). Resta in sospeso anche la questione delle categorie inconsistenti.

Per risolvere queste problematiche, includiamo sia *unknown* che *sales and marketing* tra le gruoplist. Decidiamo di associare ad *unknown* tutte le *category* presenti per le quali non abbiamo informazioni. A *sales and marketing* assoceremo solo le *category* per le quali è l'unica grouplist. Ripetendo il conteggio otteniamo il risultato 3 in Figura 16.

Vediamo come il numero di entry utilizzabili come training set sia aumentato significativamente: da circa 30 mila a circa 40 mila. *Advertising* è ancora la categoria più gettonata, *sales and marketing* e *unknown* sono entrambe presenti in quantità

significativa. Il miglioramento è notevole anche nella distribuzione dei label. Appare evidente che nei casi precedenti non stavamo gestendo correttamente alcune casistiche. Poiché non emergono altre problematiche, possiamo dirci soddisfatti di questo training set.

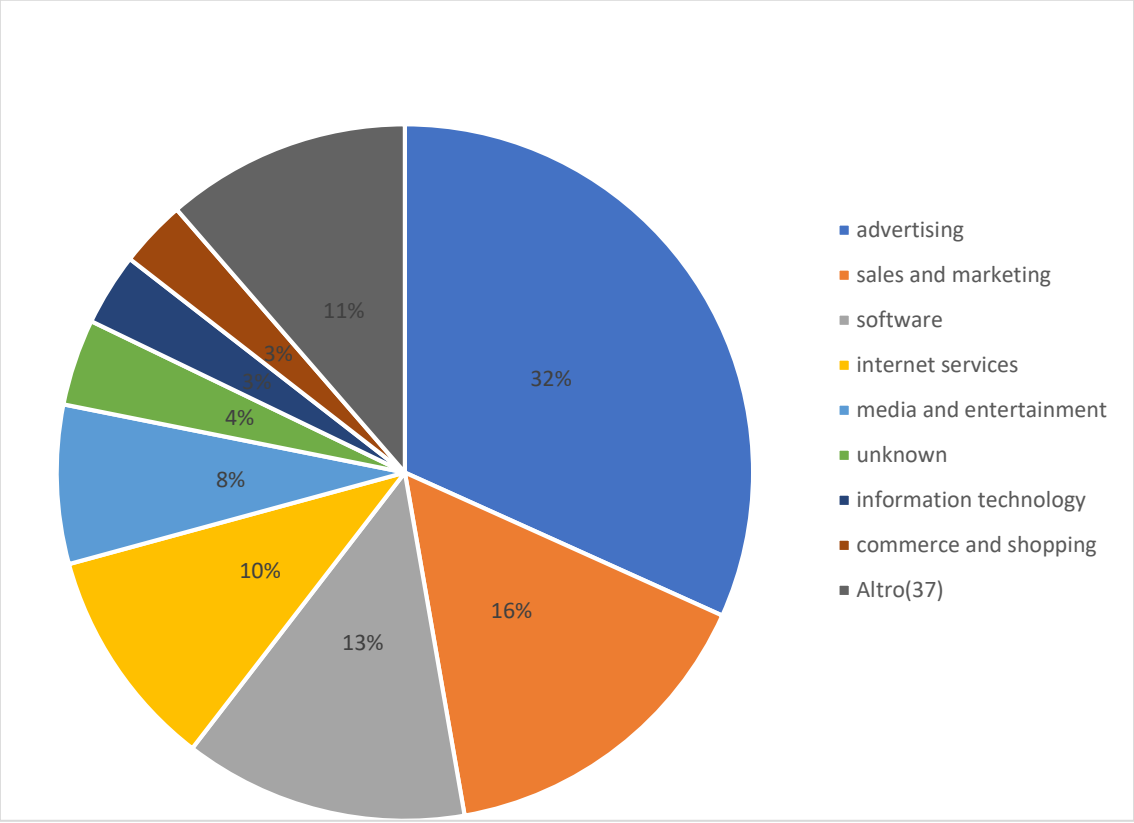


Figura 15: Distribuzione delle categorie nel training set

Procediamo con l'allenamento della rete. Tenteremo diverse architetture utilizzando il solver *'sgd' (stochastic gradient descent)*. Useremo 20000 campioni del dataset per allenare il modello e tutti gli altri per valutarne le performance. Gli altri parametri forniti alla rete sono i seguenti:

- *max_iter=300*
- *alpha=1e-4*
- *tol=1e-4*
- *learning_rate_init=.1*

Riportiamo i risultati (media di tre tentativi) per le varie architetture scelte.

Architettura	Score medio
10	0.864
50	0.924
100	0.929
50 - 50	0,931
50 - 100	0.930

Tabella 11: Score delle architetture per la rete neurale

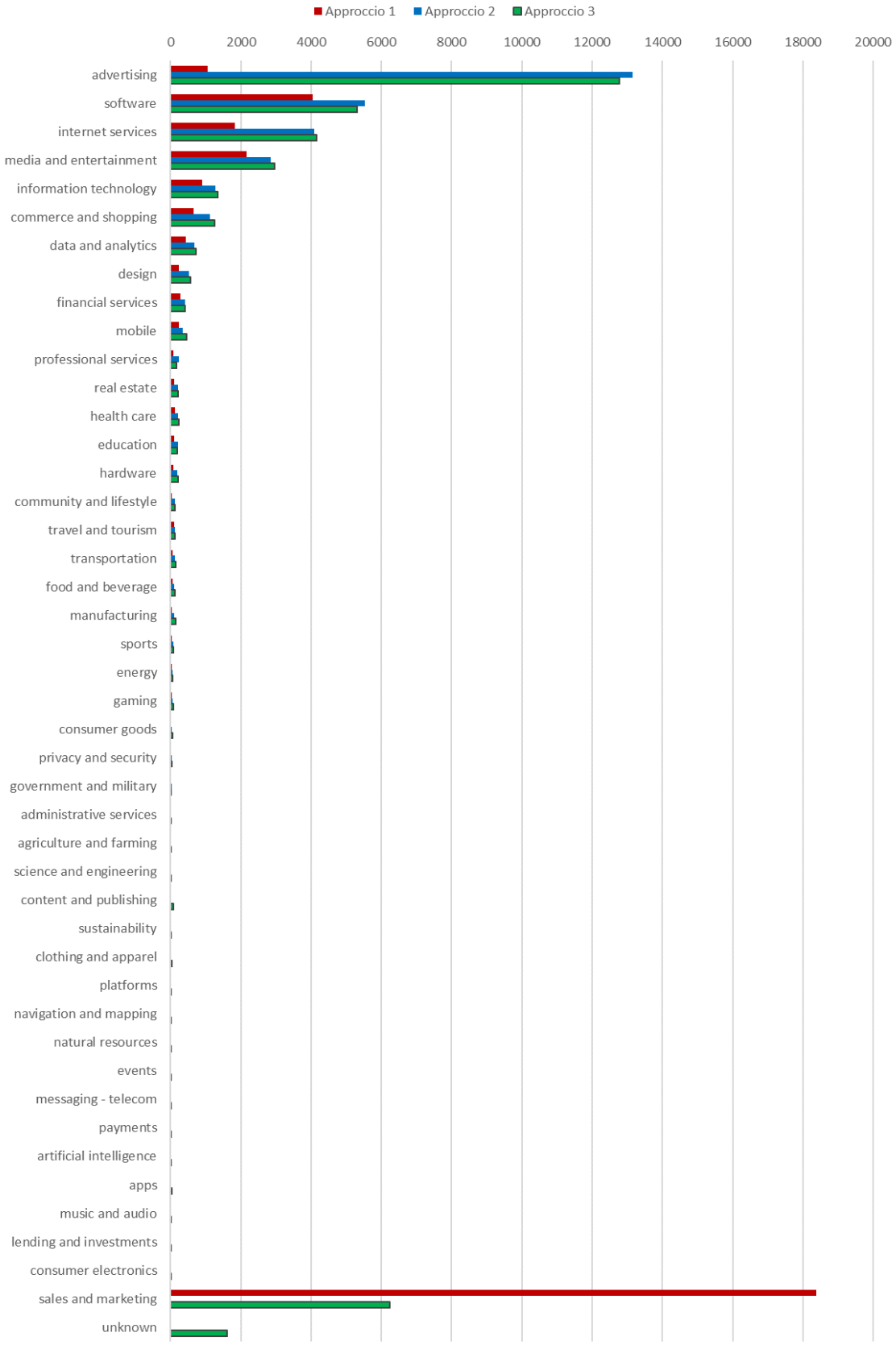


Figura 16: Distribuzione delle categorie per vari approcci

L’analisi rileva che la miglior architettura è la 50-50 anche se, esclusa quella a 10 nodi i quali sono evidentemente insufficienti, la differenza è leggerissima rispetto alle altre architetture testate. La rete allenata è precisa circa al 93%, un punteggio altissimo.

Normalmente potremmo considerarci soddisfatti ma vale la pena osservare il training error. È prossimo allo zero, talvolta in alcune run è esattamente zero. Questo è, di solito, un segnale di overfitting, la rete sta imparando troppo bene il dataset fornitole.

Verifichiamo quindi il comportamento della rete con i dati incerti. Ricordiamo che vorremmo la rete scegliesse per noi una delle grouplist presente nell’elenco. In caso di scelta di una grouplist non presente, avremo un netto fallimento. Considereremo accettabile invece, la scelta di qualsiasi grouplist presente, non strettamente una delle due o più a parimerito con la maggior cardinalità.

Riportiamo i risultati di tre diverse run con architettura 50-50. Per ogni run eseguiremo uno shuffle dei dati, in modo da permutare leggermente training e test set.

Training error	Test error	Precisione	Tot corrette	Tot errate
0.001	0.057	98.74%	21656	275
0.000	0.057	99.09%	21732	199
0.001	0.063	98.74%	21655	276

Tabella 12: Prestazioni della rete neurale a confronto

I risultati sono complessivamente ottimi: la rete seleziona quasi sempre una grouplist coerente. Il modello allenato potrà essere utilizzato per la scelta della grouplist nei prossimi esperimenti. In generale, possiamo considerarlo un’interessante alternativa ad una scelta puramente casuale.

6.2 Clustering dei dati

Eseguiamo il primo clustering sul dataset utilizzando il dataset originale. Setteremo il parametro *n_init = 3*. Questo parametro indica quante volte, per un dato valore di k, k-means ripeterà l'esecuzione dell'algoritmo a partire da nuovi centroidi. Osserviamo in Figura 17 la curva di Elbow per visualizzare il possibile punto di ginocchio.

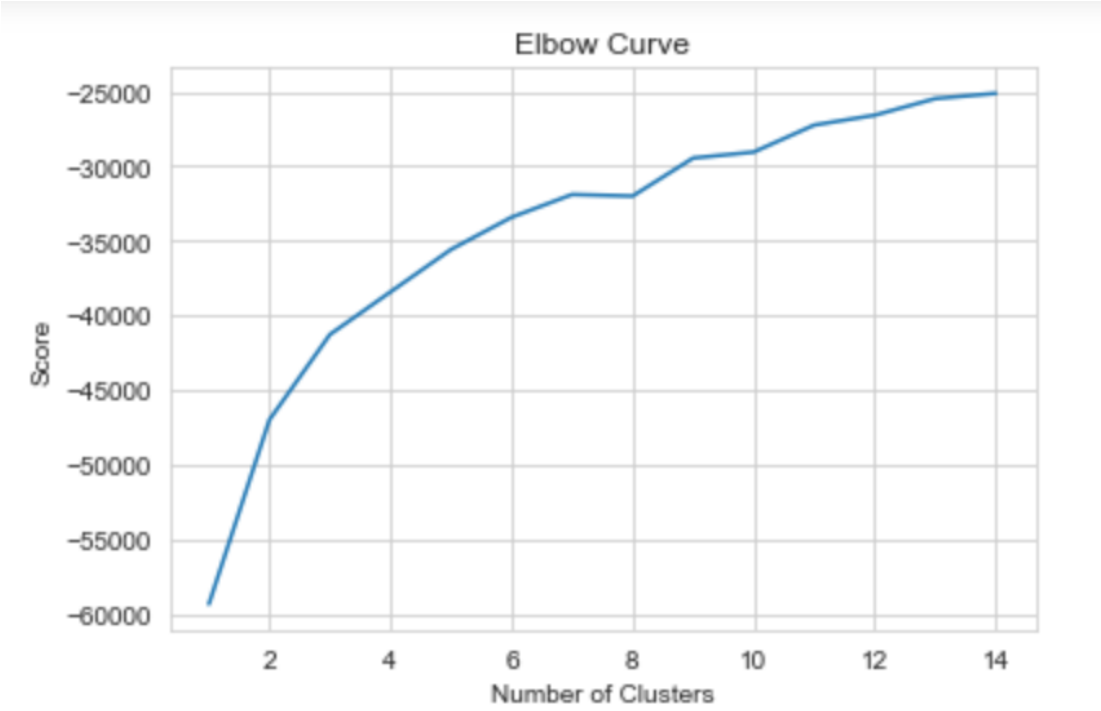


Figura 17: Elbow curve per dataset completo

Purtroppo, la curva non è buona, il ginocchio non è facilmente individuabile. Andiamo a calcolare il coefficiente di silhouette per diversi valori crescenti di k. Riportiamo in Tabella 13 i risultati

Numero *k* di cluster	Coefficiente di silhouette
2	0.211
3	0.258
4	0.214
5	0.218
10	0.276
15	0.335
25	0.441
40	0.475

Tabella 13: valori di silhouette per k crescente

I risultati non sono incoraggianti. Usare un grande numero di cluster induce artificialmente il coefficiente di silhouette ad aumentare, ma individua insiemi di dati sempre meno rilevanti.

Ripetiamo l'analisi per il dataset a cui abbiamo aggiunto le feature burn rate, variazione di burn rate e investimenti ripetuti.

Anche in questo caso, la curva di Elbow, Figura 18, non rileva un punto di ginocchio.

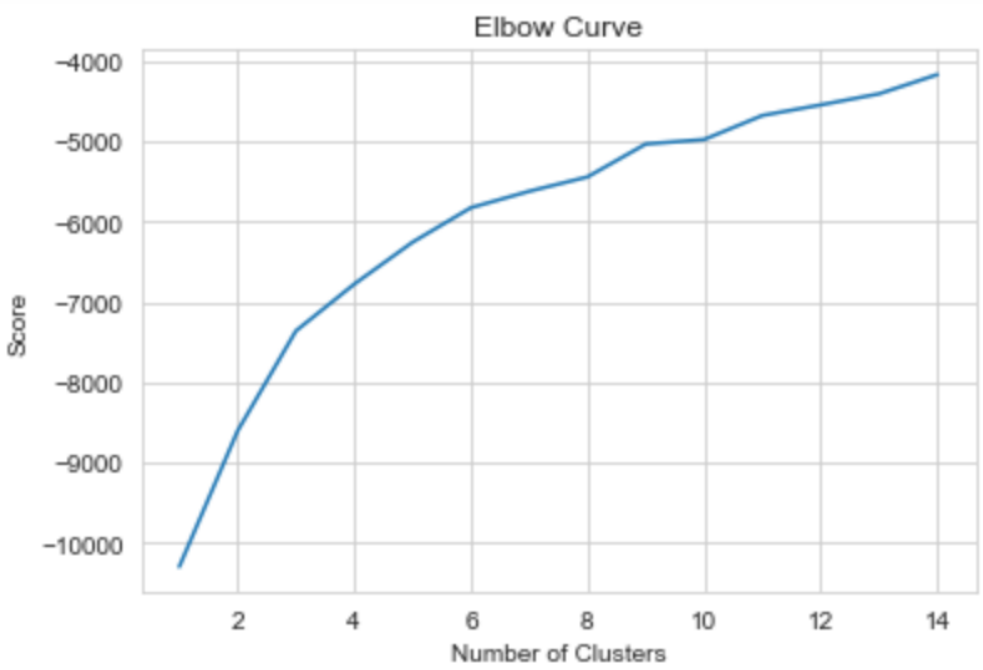


Figura 18: elbow curve per dataset arricchito

Osserviamo i nuovi coefficienti di silhouette in Tabella 14.

Numero *k* di cluster	**Coefficiente di silhouette**
2	0.160
3	0.223
4	0.215
5	0.187
10	0.262
15	0.333
25	0.349
40	0.461

Tabella 14: coefficienti di silhouette per dataset arricchito

I valori sono complessivamente più bassi, segno che l'informazione aggiuntiva del numero di investimenti e del totale ricevuto non da contributo particolarmente significativo.

6.3 Support Vector Machine (SVM)

Nei nostri esperimenti con SVM, utilizzeremo il metodo di scikit learn GridSearchCV (sklearn.org, s.d.).

Come quasi tutti gli algoritmi di machine learning, SVM è un algoritmo che riceve in input una sequenza di parametri che ne influenzano l'output. Per individuare quelli ottimali da utilizzare per il nostro caso specifico, utilizziamo GridSearchCV, con parametro di cross-validation *cv*=5.

Il metodo suddivide il training set in 5 parti e lo usa per testare e confrontare il comportamento del modello al variare dei parametri. Utilizzeremo due kernel: lineare e RBF (radial basis function). Il kernel trick è comunemente usato in SVM. Prenderemo in esame per entrambi i kernel il parametro 'C', associato alla penalità che il modello assegna ai campioni classificati non correttamente. Aumentare 'C' comporta la scelta di un iperpiano con margine inferiore, ma che classifica correttamente più punti possibile. Diminuendolo, SVM prediligerà iperpiani con margine maggiore, a discapito di un training error maggiore.

Per il kernel RBF, il parametro gamma definisce la portata dell'influenza di un singolo campione del training set, con valori bassi che significano "lontano" e valori alti che significano "vicino". Il comportamento del modello è molto sensibile al parametro gamma. Se gamma è troppo grande, il raggio dell'area di influenza dei vettori di supporto finisce con l'includere solo il vettore di supporto stesso e nessuna quantità di regolarizzazione C sarà in grado di prevenire l'overfitting.

Selezioneremo, per entrambi i kernel, il parametro o la combinazione di parametri che fornisce lo score più alto, ed utilizzeremo questi per allenare il modello.

Iniziamo ad allenare il modello usando tutte le organizzazioni in stato *exit* e *closed* disponibili. Per questo primo test utilizzeremo solo le informazioni provenienti dal dataset non arricchito. Poiché, come mostrato in precedenza, le organizzazioni in stato *closed* (1560) sono significativamente meno di quelle in *exit* (3975) selezioneremo casualmente un sottoinsieme di queste in modo che abbia cardinalità sarà pari al numero di *closed.* Utilizzeremo 2000 campioni (1000 *exit* e 1000 *closed*) per l'allenamento del modello, e 1120 campioni (560 *exit* e 560 *closed*) per la valutazione delle prestazioni. Il dataset simmetrico rende molto più semplice il processo di lettura ed interpretazione dei dati. Prendere un sottoinsieme dei dati è un'operazione legittima, non ne altera infatti la distribuzione spaziale oggetto del nostro studio.

Riportiamo i risultati per i due kernel.

Linear				**RBF**			
Parametri		'C': 1		**Parametri**		'C': 1 'gamma': 1.0	
Accuracy		0.723		**Accuracy**		0.724	
Precision				**Precision**			
Closed		0,686		**Closed**		0,727	
Exit		0,777		**Exit**		0,721	
Recall				**Recall**			
Closed		0,821		**Closed**		0,718	
Exit		0,623		**Exit**		0,730	
F1-score				**F1-score**			
Closed		0,747		**Closed**		0,722	
Exit		0,692		**Exit**		0,726	
Confusion matrix				**Confusion matrix**			
		Predicted				**Predicted**	
		Closed	Exit			Closed	Exit
Actual	Closed	349	211	**Actual**	Closed	409	151
	Exit	100	460		Exit	158	402

Tabella 15: Risultati a confronto, dataset originale

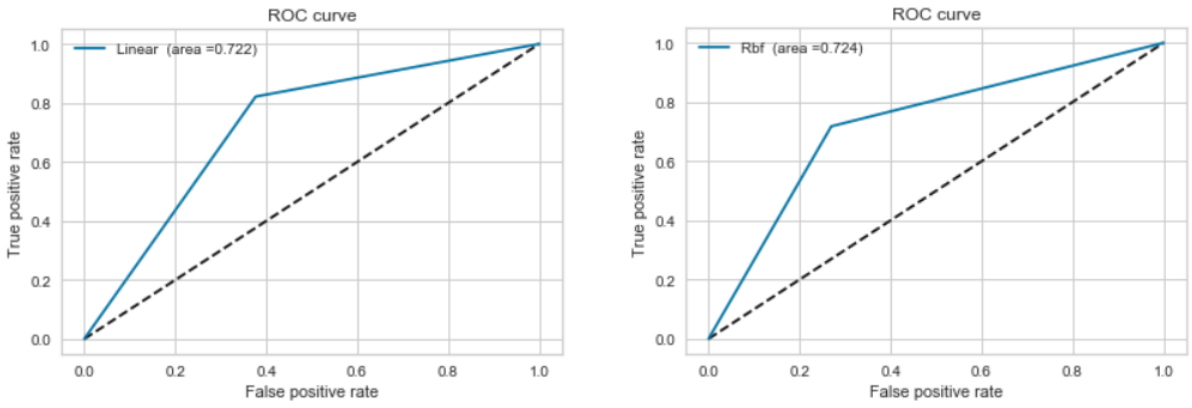


Figura 19: ROC curve a confronto

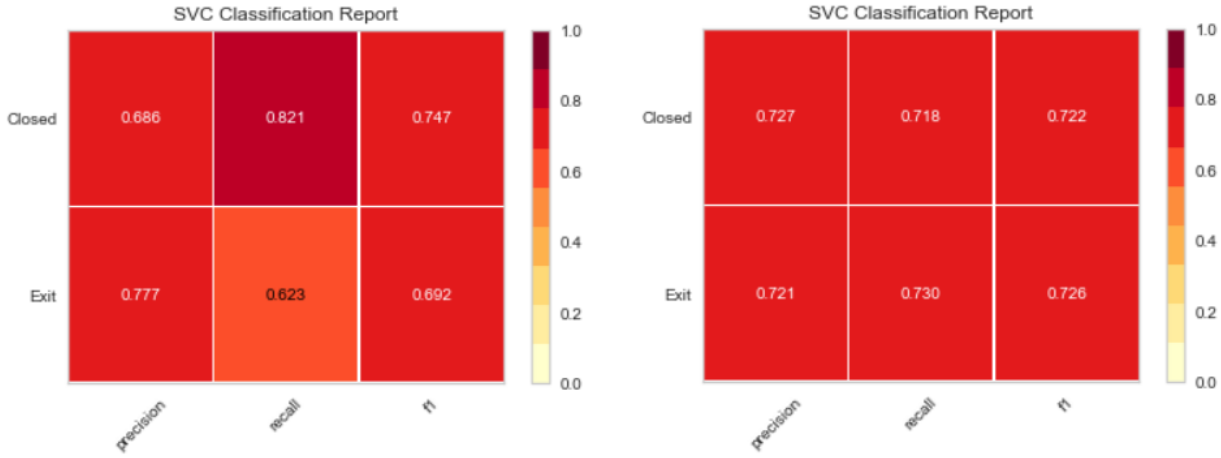


Figura 20: Linear (a destra) e RBF (a sinistra) a confronto

Utilizzeremo ora il dataset arricchito dalle colonne *burn rate*, *accelerazione* e *fiducia*. Riportiamo i risultati dell'analisi in Tabella 16.

Linear				**RBF**			
Parametri		'C': 1		**Parametri**		'C': 1 'gamma': 1.0	
Accuracy		0.761		**Accuracy**		0.748	
Precision				**Precision**			
Closed		0,744		**Closed**		0,762	
Exit		0,780		**Exit**		0,736	
Recall				**Recall**			
Closed		0,795		**Closed**		0,721	
Exit		0,727		**Exit**		0,775	
F1-score				**F1-score**			
Closed		0,769		**Closed**		0,741	
Exit		0752		**Exit**		0,755	
Confusion matrix				**Confusion matrix**			
		Predicted				**Predicted**	
		Closed	Exit			Closed	Exit
Actual	Closed	407	153	**Actual**	Closed	434	126
	Exit	115	445		Exit	156	404

Tabella 16: Risultati a confronto, dataset arricchito

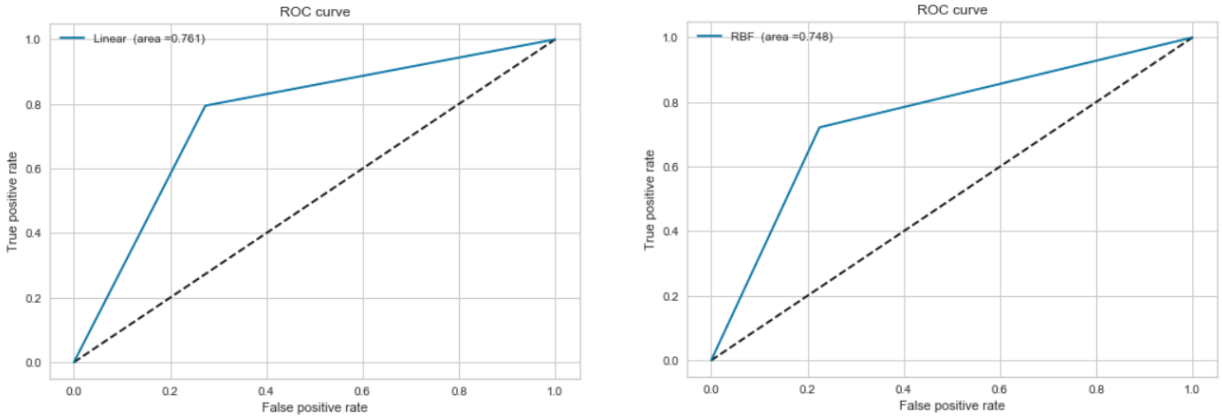


Figura 21: ROC curve a confronto

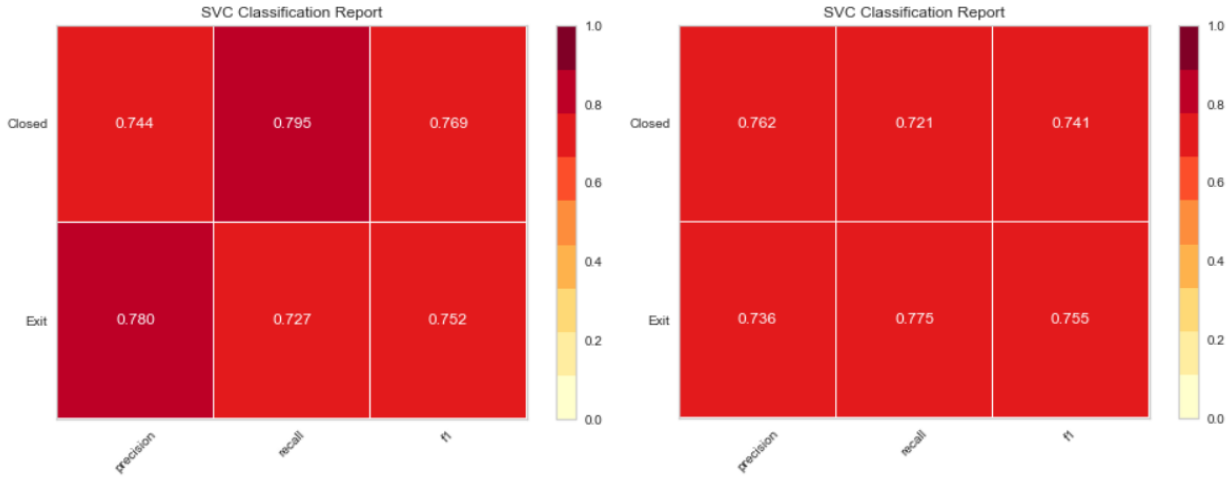


Figura 22: Linear (a destra) e RBF (a sinistra) a confronto

6.4 Neural Network

Utilizziamo gli stessi dataset per allenare una rete neurale. Tenteremo diverse architetture ed utilizzeremo ancora una volta GridSearchCV per individuare la migliore. I parametri che forniremo in ingresso all'algoritmo sono i seguenti:

max_iter=300, alpha=0,0001, solver='sgd', tol=0,0001

max_iter=1000, alpha=0,0001, solver='adam', tol=0,0001

Ricordiamo che la rete inizia inizializzando casualmente i valori dei pesi associati agli archi che collegano i nodi.

Il primo parametro *max_iter* fornisce un limite superiore al numero di iterazioni in cui la rete corregge i pesi dei nodi per classificare correttamente i dati. Per il solver *adam* sono utili più iterazioni.

alpha è un parametro di regolarizzazione: serve ad evitare l'*overfitting* penalizzando i lati con pesi troppo elevati.

Il *solver* indica la formula utilizzata per l'assegnamento dei pesi. '*sgd'* ed *'adam'* sono due versioni alternative di *stochastic gradient descent.*

tol è l'abbreviazione di *tolerance*, intesa come tolleranza per l'ottimizzazione. Quando la perdita o lo *score* non migliorano di almeno *tol* per un certo numero di iterazioni consecutive (fissato di default a 10), la convergenza è considerata raggiunta e l'allenamento si ferma.

Le seguenti architetture sono state testate:

(10,) (50,) (100,) (10,50) (50,50,) (50,25,10)

Riportiamo i risultati dell'analisi per il dataset originale in Tabella 17 e per il dataset arricchito in Tabella 18.

Solver	Miglior architettura	Score	Training error	Test error
sgd	(50,)	0.738	0.232	0.246
adam	(10,)	0.735	0.213	0.288

Tabella 17: Risultati per Rete Neurale su dataset originale

Solver	Miglior architettura	Score	Training error	Test error
sgd	(10, 50)	0.743	0.226	0.266
adam	(50,)	0.703	0.135	0.315

Tabella 18: Risultati per Rete Neurale su dataset arricchito

Decidiamo di utilizzare *sgd* con le migliori architetture individuate. Riportiamo i risultati per il dataset originale e per il dataset arricchito a confronto.

Dataset Originale				Dataset Arricchito			
Accuracy		0.712		**Accuracy**		0.734	
Precision				**Precision**			
Closed		0,732		**Closed**		0,715	
Exit		0,695		**Exit**		0,757	
Recall				**Recall**			
Closed		0,668		**Closed**		0,779	
Exit		0,755		**Exit**		0,689	
F1-score				**F1-score**			
Closed		0,698		**Closed**		0,745	
Exit		0,724		**Exit**		0,721	
Confusion matrix				**Confusion matrix**			
		Predicted				**Predicted**	
		Closed	Exit			Closed	Exit
Actual	Closed	410	150	**Actual**	Closed	386	174
	Exit	126	434		Exit	124	436

Tabella 19: Risultati a confronto Neural Network

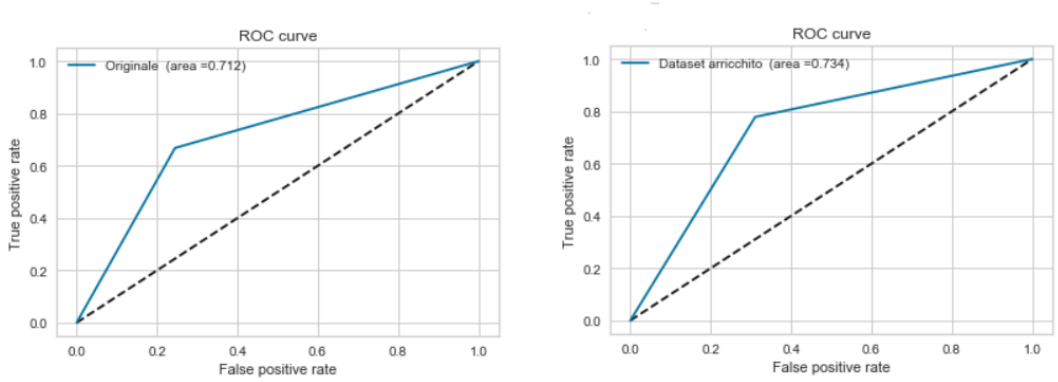


Figura 23: ROC curve a confronto Neural Network

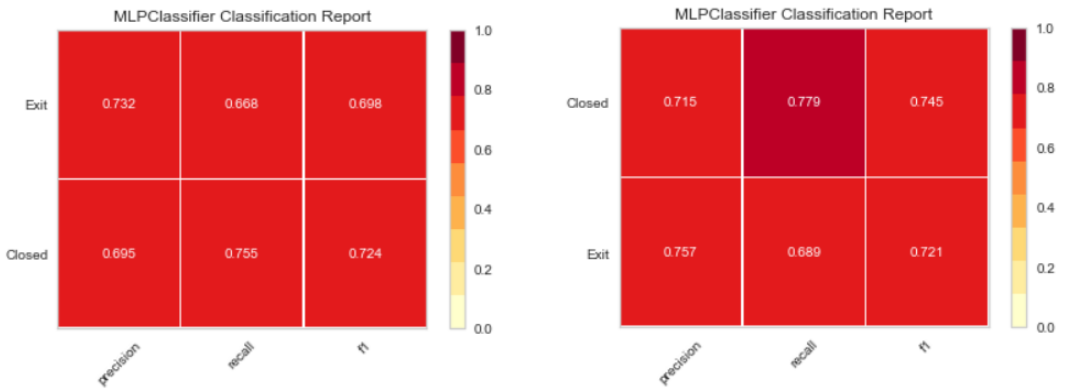


Figura 24: Classification report a confronto Neural Network

6.5 Adaboost Random Forest

Per l'algoritmo Random Forest valuteremo i seguenti parametri:

- base_estimator__criterion : ["gini", "entropy"],
- base_estimator__splitter : ["best", "random"],
- n_estimators: [200, 500, 700]

Riportiamo i risultati dell'analisi sui parametri.

Dataset	Best criterion	Best splitter	n-estimators	Train score	Accuracy
Originale	Balanced	Entropy	500	0.992	0.733
Arricchito	Balanced	Gini	700	0.997	0.715

Tabella 20: Migliori parametri stimati per Random Forest

E il confronto tra i due dataset.

Dataset Originale				**Dataset Arricchito**			
Accuracy		0.761		**Accuracy**		0.734	
Precision				**Precision**			
Closed		0,732		**Closed**		0,709	
Exit		0,736		**Exit**		0,722	
Recall				**Recall**			
Closed		0,738		**Closed**		0,730	
Exit		0,730		**Exit**		0,700	
F1-score				**F1-score**			
Closed		0,735		**Closed**		0,719	
Exit		0733		**Exit**		0,711	
Confusion matrix				**Confusion matrix**			
		Predicted				**Predicted**	
		Closed	Exit			Closed	Exit
Actual	Closed	409	151	**Actual**	Closed	392	168
	Exit	147	413		Exit	151	409

Tabella 21:Risultati a confronto Random Forest

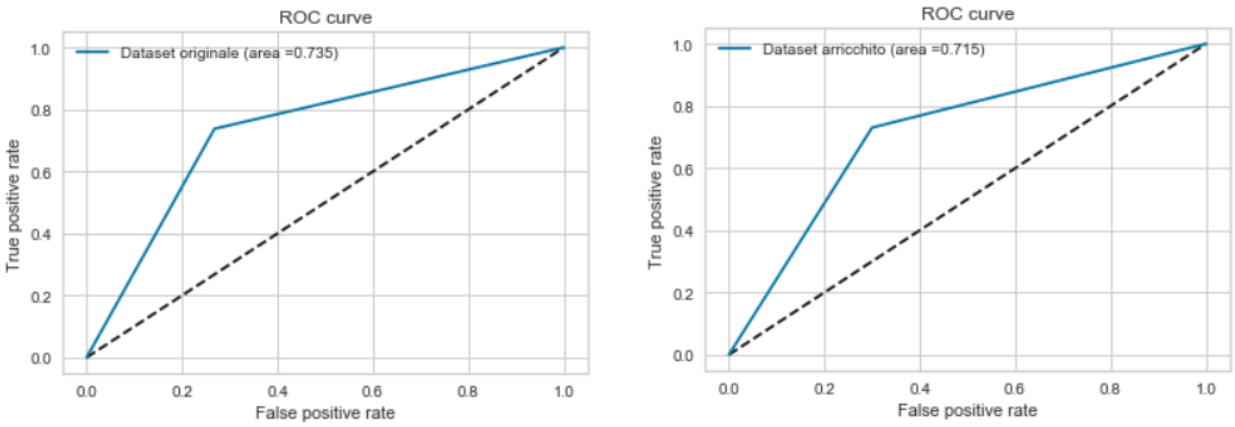


Figura 25: ROC curve a confronto, Random Forest

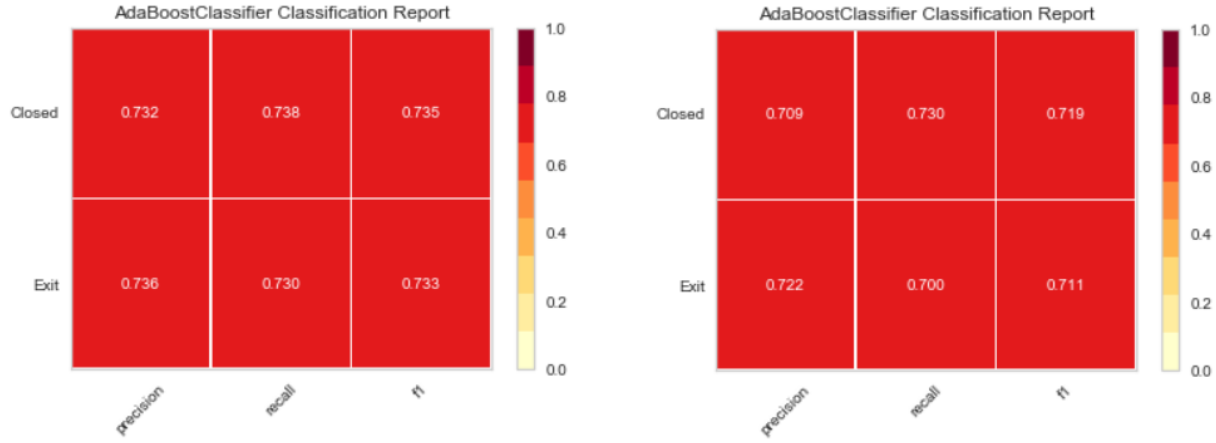


Figura 26: Classification report, Random Forest

7 Analisi dei risultati

Con l'esclusione del clustering, la misura di valutazione a cui siamo maggiormente interessati è l'*accuracy score* o *test error*, il quale è statisticamente molto simile a quello che noi stimiamo essere il vero errore medio commesso dal classificatore nell'etichettare nuovi campioni.

7.1.1 Clustering

Utilizzando k-means non siamo riusciti ad individuare un valore di *k* utile a produrre un clustering che individuasse strutture significative. Questo risultato negativo non fornisce tuttavia argomenti sufficienti ad escludere la possibilità di clusterizzare efficacemente i dati. Una possibile strategia potrebbe consistere nell'applicare una trasformazione non lineare allo spazio vettoriale definito dal dataset. È una tecnica concettualmente identica al *kernel trick* utilizzato per SVM.

7.1.2 SVM

Il classificatore SVM in tutte le sue varianti continua ad avere un errore oscillante intorno al 25%. Da questo possiamo trarre due conclusioni.

La prima, molto positiva, è che il classificatore ha effettivamente imparato qualcosa sui dati. In caso contrario avrebbe un errore prossimo al 50%, in quanto per ogni entry sceglierebbe un valore sostanzialmente casuale. Inoltre, poiché il training error è molto simile al test error, non siamo in una situazione di *overfitting*, nel cui caso avremmo un errore molto più basso sul training set rispetto al test set; in altre parole il modello imparerebbe troppo bene l'insieme di test, ma non la reale distribuzione dei dati.

La seconda, è che il nostro modello non sta catturando correttamente alcuni aspetti dei dati. Una possibile spiegazione di ciò è che le scelte del modello e della sua complessità non sono adatte a questo task. In questo caso si parla di *underfitting*, si veda la Figura 27, il modello generalizza troppo.

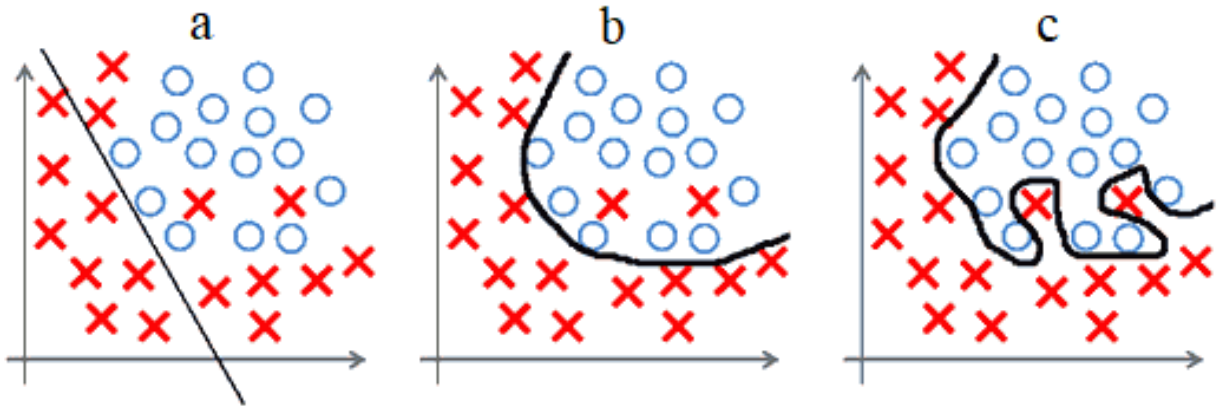


Figura 27: Confronto tra: a) underfitting, b) fitting, c) overfitting

Il dataset arricchito non ha dato i risultati sperati. L'obiettivo dell'aggiunta delle variabili era catturare quelle caratteristiche del dataset che sfuggivano alla prima analisi. I risultati evidenziano come in media l'incremento dell'accuracy sia presente, ma molto modesto. Possiamo argomentare che il beneficio portato dalla nuova informazione aggiunta non è sufficiente a compensare l'aumento della complessità del modello da elaborare. Talvolta, le prestazioni sono leggermente inferiori rispetto al caso più semplice.

Dalla *confusion matrix* vediamo che SVM è leggermente sbilanciato e tende a classificare molte più aziende come *exit*, si veda la recall in Figura 20, questo induce a pensare che per migliorare le prestazioni di questo modello potrebbe essere utile aggiungere feature che evidenzino maggiormente i trend negativi per le imprese.

7.1.3 Rete neurale

La rete neurale ottiene mediamente prestazioni leggermente inferiori rispetto ad SVM. La scelta dell'architettura generalmente ricade su quelle con un solo *hidden layer*. La rete utilizza quindi un iperpiano per separare i dati invece di una struttura più complessa. Questo può dipendere anche dalla dimensione ridotta del dataset. Le reti neurali sono particolarmente esigenti in termini di mole di dati, pertanto il potenziale di questo algoritmo non sta venendo sfruttato appieno. Qualora divenissero disponibili nuovi campioni, è tra tutti quelli analizzati l'algoritmo con i maggiori margini di miglioramento.

Anche in questo caso, il beneficio della nuova informazione nel dataset arricchito è mediamente presente, ma marginale.

Il solver *sgd* si rivela più efficace di *adam*. Il secondo è più esigente in termini di numero di iterazioni per raggiungere la convergenza, ma il risultato conduce più spesso ad *overfitting* dei dati che ad un effettivo miglioramento del modello. Questo è più evidente nel dataset arricchito, osserviamo in Tabella 18 come presenti training error molto più basso del test error. Ci troviamo nel caso c) della Figura 27.

Nel corso di molteplici esperimenti, la rete neurale sembra più orientata verso l'assegnazione dei label closed, al contrario di SVM.

7.1.4 Random Forest

La random forest ha prestazioni talmente simili ad SVM che è difficile decidere quale dei due sia più performante. Questo algoritmo è conosciuto per essere particolarmente efficace anche in presenza di dataset ridotti. Considerato che ancora una volta le prestazioni restano sostanzialmente invariate, si rafforza sempre di più la tesi secondo cui non sono stati individuati alcuni fattori discriminanti necessari per migliorare ulteriormente le prestazioni degli algoritmi.

Un punto a favore della random forest è avere la confusion matrix più bilanciata. In altre parole, a differenza degli altri modelli questo non sembra presentare *bias*. In Figura 28 osserviamo infine come si presenta il decision tree per il nostro problema decisionale.

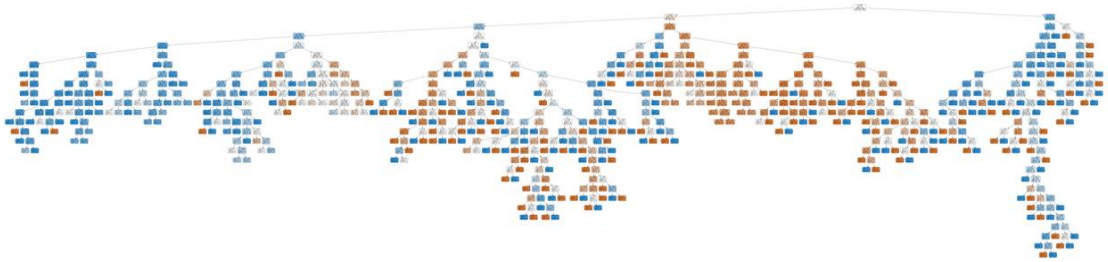


Figura 28: Decision tree prodotto da Random Forest

8 Conclusioni e sviluppi futuri

Abbiamo verificato la presenza di un grande potenziale nelle informazioni raccolte da Crunchbase. Gli algoritmi da noi testati hanno tutti ottenuto un successo significativo anche se parziale. Riusciamo a classificare correttamente il 75% dei campioni: questo induce alla conclusione che, nell'attuale implementazione, i modelli stiano facendo underfitting dei dati.

Esiste la possibilità per cui il fenomeno che stiamo studiando contenga in sé una componente randomica intrinseca. In questo caso, anche con l'allenamento di modelli più complessi su di un dataset più ricco, il valore della percentuale di accuratezza si manterrebbe prossimo a quello da noi misurato.

L'analisi dei valori di precision e recall rilevano un comportamento leggermente diverso dei tre algoritmi. Nelle label assegnati da SVM osserviamo la maggior concentrazione di falsi positivi, mentre in quelle predette dalla Rete Neurale, di falsi negativi. Nell' output di Random Forest, invece, non si sono osservati sbilanciamenti significativi.

Questo non implica necessariamente che il modello Random Forest sia sempre da preferire. La Rete Neurale è un modello conservativo: in caso di incertezza, propendendo per il valore "closed", induce a non mettere il capitale a rischio. Al contrario, SVM è un modello che spinge verso gli investimenti, contestualmente più adatto a business angels o a chi cerca margini di guadagno più elevati, al prezzo di una minor sicurezza.

Anche se ci sentiamo di escludere che un algoritmo potrà mai fornire una previsione affidabile sul destino di una singola azienda, è possibile che modelli come quelli studiati in questo elaborato possano un domani essere effettivamente un utile tool di supporto per gli investitori, oppure utilizzati per fare analisi ad ampio spettro, magari a livello regionale o per settore.

Limitazioni

La principale limitazione riscontrata deriva dall'espressività limitata delle feature disponibili e di quelle finora implementate. Questo evidenzia l'importanza della fase di feature engineering. I valori calcolati di *burn rate,* della sua variazione e la ripetizione degli investimenti, intuitivamente dovrebbero dare indicazioni preziose sull'attività di un'impresa e sull'interesse che stimola negli investitori, tuttavia il beneficio apportato ai modelli è molto limitato.

Sviluppi futuri

Gli esperimenti da noi condotti si possono ripetere apportando modifiche ed aggiornamenti, allo scopo di ottenere modelli ancora più performanti.

Si può esplorare ulteriormente il dataset, potrebbe essere interessante selezionare un'altra tecnologia ed un altro campo di applicazione per verificare se in quel caso i classificatori hanno prestazioni affini.

Si può ridefinire il concetto di successo per un'azienda, non limitandosi alla dicotomia exit/closed. Si può tentare ad esempio un multiclassificatore a più label, oppure si potrebbero utilizzare algoritmi di *regressione.* Questi possono essere adatti per prevedere la data di exit o di closing per un'impresa, come anche la data e/o l'ammontare del successivo funding round.

Infine, si potrebbero esplorare l'applicazione del deep learning al presente problema. Tuttavia, tale approccio richiederebbe un aumento sostanziale del numero di campioni considerati.

9 Indice delle figure

Figura 1: Diagramma di Gartner 10
Figura 2: Numero di imprese fondate per anno, distinte per settori 12
Figura 3: Crunchbase, logo ufficiale. © 2019 Crunchbase Inc. 17
Figura 4: output del metodo info() della libreria pandas 28
Figura 5: output del metodo matrix() della libreria missingno 28
Figura 6: dataset risultante dopo l'eliminazione dei valori NA 33
Figura 7: Funding round per impresa 37
Figura 8: Funding round per impresa 37
Figura 9: Esempio con 3 cluster di punti 41
Figura 10: Curva di Elbow ideale. In rosso il punto di ginocchio 43
Figura 11: Esempi di iperpiani separatori, l'algoritmo cerca di selezionare quello che meglio classifica i dati 44
Figura 12: Esempio di decision tree 45
Figura 13: Schema di una rete neurale 47
Figura 14: esempi di espressività di una rete neurale con uno (a), due (b) e tre (c) hidden layer. 48
Figura 15: Distribuzione delle categorie nel training set 52
Figura 16: Distribuzione delle categorie per vari approcci 53
Figura 17: Elbow curve per dataset completo 55
Figura 18: elbow curve per dataset arricchito 56
Figura 19: ROC curve a confronto 58
Figura 20: Linear (a destra) e RBF (a sinistra) a confronto 59
Figura 21: ROC curve a confronto 60
Figura 22: Linear (a destra) e RBF (a sinistra) a confronto 60
Figura 23: ROC curve a confronto Neural Network 62
Figura 24: Classification report a confronto Neural Network 62
Figura 25: ROC curve a confronto, Random Forest 64
Figura 26: Classification report, Random Forest 64
Figura 27: Confronto tra: a) underfitting, b) fitting, c) overfitting 65
Figura 28: Decision tree prodotto da Random Forest 67

10 Indice delle tabelle

Tabella 1: Category_groups 19
Tabella 2: Acquisitions 20
Tabella 3: Funding Rounds 21
Tabella 4: Companies 22
Tabella 5: Investors 23
Tabella 6: Investments 24
Tabella 7: IPOS 25
Tabella 8: esempi di categoryList e categoryGroupList estratto da companies 32
Tabella 9: Group list associate alle category 32
Tabella 10: Distribuzione per status e investimenti ricevuti 34
Tabella 11: Score delle architetture per la rete neurale 52
Tabella 12: Prestazioni della rete neurale a confronto 54
Tabella 13: valori di silhouette per k crescente 55
Tabella 14: coefficienti di silhouette per dataset arricchito 56
Tabella 15: Risultati a confronto, dataset originale 58
Tabella 16: Risultati a confronto, dataset arricchito 59
Tabella 17: Risultati per Rete Neurale su dataset originale 61
Tabella 18: Risultati per Rete Neurale su dataset arricchito 61
Tabella 19: Risultati a confronto Neural Network 62
Tabella 20: Migliori parametri stimati per Random Forest 63
Tabella 21:Risultati a confronto Random Forest 63

11 Riferimenti Bibliografici

Amazon inc. (s.d.). *Amazon Alexa*. Tratto il giorno Settembre 30, 2019 da https://developer.amazon.com/it/alexa

Apple inc. (s.d.). *Siri* . Tratto il giorno Settembre 30, 2019 da Apple (IT): https://www.apple.com/it/siri/

Baldacci, O. (2019, Aprile 1). Volere la luna non è più un'utopia. *L'Osservatore Romano*. Tratto il giorno Settembre 30, 2019 da http://www.osservatoreromano.va/it/news/volere-la-luna-non-e-piu-unutopia

BATISTA, F., & CARVALHO, J. P. (2015). Text based classification of companies in CrunchBase. In IEEE (A cura di), *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, (p. 1 - 7).

Bendinelli, S. (2016, Settembre 19). *L'aumento dei prezzi di Uber durante l'esplosione a New York.* Tratto il giorno Ottobre 4, 2019 da the Submarine: https://thesubmarine.it/2016/09/19/uber-new-york-esplosione-prezzi/

BMW. (2019, Luglio 10). *I cinque passaggi della guida autonoma*. Tratto il giorno Settembre 30, 2019 da BMW.com: https://www.bmw.com/it/automotive-life/guida-autonoma.html

BrandTotal. (s.d.). *BrandTotal | Real Time Competitive Analysis*. Tratto il giorno Ottobre 2, 2019 da https://www.brandtotal.com/

Brilliant.org. (s.d.). *Social Networks*. Tratto il giorno Agosto 16, 2019 da https://brilliant.org/wiki/social-networks/

BusinessDictionary.com. (2019, Settembre). *What is a lead investor? definition and meaning.* Tratto il giorno Settembre 10, 2019 da http://www.businessdictionary.com/definition/lead-investor.html

Chatbots.org. (s.d.). *Italian Chatbots Directory*. Tratto il giorno Settembre 30, 2019 da Chatbots.org: https://chatbots.org/language/italian/

CHERKASSKY, V., & MULIER, F. M. (2007). Introduction. In *Learning from data: concepts, theory, and methods* (p. 3 - 4). John Wiley & Sons.

community, T. S. (A cura di). (2017, Giugno 10). *Data types NumpY v1.13 Manual*. Tratto il giorno Settembre 6, 2019 da ScyPy.org: https://docs.scipy.org/doc/numpy-1.13.0/user/basics.types.html

Conversation Intelligence for Sales Teams. (s.d.). Tratto il giorno Ottobre 2, 2019 da Chorus.ai: https://www.chorus.ai/?

CORMEN, T. H., & al., e. (2009). *Introduction to algorithms.* MIT press.

Crunchbase Inc. (s.d.). *About Crunchbase · What Is Crunchbase? · Identify New Companies.* Tratto il giorno Settembre 30, 2019 da Crunchbase: https://about.crunchbase.com/about-us/

Crunchbase inc. (s.d.). *Funding Rounds | Crunchbase.* Tratto il giorno Ottobre 4, 2019 da Crunchbase: https://www.crunchbase.com/organization/crunchbase#section-funding-rounds

Excelle S.r.l. (s.d.). *Next Best Action Analytics : cos'è il marketing contestuale*. Tratto il giorno Settembre 15, 2019 da Excelle: http://www.excelle.it/next-best-action-analytics/

Fedorov, S. (2019, Febbraio 19). *The Offering Price vs. the Opening Price of an IPO.* Tratto il giorno Settembre 10, 2019 da Zacks.com: https://finance.zacks.com/offering-price-vs-opening-price-ipo-2670.html

Gartner. (2019, Settembre). *Hype Cycle*. Tratto il giorno Settembre 24, 2019 da gartner.com: https://www.gartner.com/en/information-technology/research/hype-cycle

Georgiou, A. (2018, Novembre 6). Russia Is Going to Build a Moon Base Manned by Avatar Robots: Report. *Newsweek*. Tratto il giorno Settembre 18, 2019 da https://www.newsweek.com/russia-going-build-moon-base-manned-avatar-robots-report-1203865

Google inc. (s.d.). *Google Now. Le informazioni giuste al momento giusto. .* Tratto il giorno Settembre 30, 2019 da Google.com: https://www.google.com/intl/it/landing/now/

Google inc. (s.d.). *Homepage | DeepMind*. Tratto il giorno Settembre 30, 2019 da Deepmind: https://deepmind.com/

Google inc. (s.d.). *Livello gratuito di GCP*. Tratto il giorno 08 27, 2019 da cloud.google.com: https://cloud.google.com/free/docs/gcp-free-tier?hl=it

Hayes, A. (2019, Giugno 3). *Stock Symbol (Ticker) Definition.* Tratto il giorno Settembre 10, 2019 da Investopedia: https://www.investopedia.com/terms/s/stocksymbol.asp

IBM. (s.d.). *IBM Watson*. Tratto il giorno Settembre 30, 2019 da https://www.ibm.com/watson

Influential. (s.d.). *Influential - The Leader in Influencer Data and Conversion*. Tratto il giorno Ottobre 2, 2019 da https://influential.co/

Kaminski, J. C. (2016). *ew Technology Assessment in Entrepreneurial Financing-Can Crowdfunding Predict Venture Capital Investments?* SSRN Working paper, RWTH Aachen. Tratto da https ://ssrn.com/abstract=2829777

Kenton, W. (2019, Agosto 18). *Burn Rate Definition.* Tratto il giorno Settembre 30, 2019 da Investopedia: https://www.investopedia.com/terms/b/burnrate.asp

KODINARIYA, T. M., & MAKWANA, P. R. (2013, Novembre). Review on determining number of Cluster in K-Means Clustering. *International Journal of Advance Research in Computer Science and Management Studies, 1*. Tratto il giorno Settembre 10, 2019 da https://s3.amazonaws.com/academia.edu.documents/34194098/V1I6-0015.pdf?response-content-disposition=inline%3B%20filename%3DReview_on_determining_number_of_Cluster.pdf&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=AKIAIWOWYYGZ2Y53UL3A%2F20190919%2Fus-e

La Stampa. (2106, Marzo 25). Tweet razzisti, Microsoft chiude il chatbot Tay. *La Stampa*. Tratto il giorno Settembre 19, 2019 da https://www.lastampa.it/tecnologia/news/2016/03/25/news/tweet-razzisti-microsoft-chiude-il-chatbot-tay-1.36583608

Lee, D. (2014, Dicembre 24). *Uber 'truly sorry' for price rise during Sydney siege.* Tratto il giorno Ottobre 4, 2018 da BBC News: https://www.bbc.com/news/technology-30595406

LIANG, Y. E., & YUAN, S.-T. D. (2016, 1 26). Predicting investor funding behavior using crunchbase social network features. *Internet Research*, 74-100.

Majaski, C. (2019, Aprile 30). *The Difference Between Pre-Money vs. Post-Money.* Tratto il giorno Settembre 10, 2019 da Investopedia: https://www.investopedia.com/ask/answers/difference-between-premoney-and-postmoney/

MARINI, M. (2019, Settembre 12). Fyodor non volerà più. Il robot umanoide russo non è adatto allo spazio. *la Repubblica*. Tratto il giorno Settembre 12, 2019 da https://www.repubblica.it/scienze/2019/09/12/news/fyodor_non_volera_piu_il_robot_umanoide_russo_non_e_adatto_allo_spazio-235825557/

Micorsoft inc. (s.d.). *Cortana | La tua intelligente assistente personale virtuale*. Tratto il giorno Settembre 30, 2019 da microsoft.com: https://www.microsoft.com/it-it/windows/cortana

Microsoft. (2017, Gennaio 22). *The Microsoft Cognitive Toolkit - CNTK | Microsoft Docs*. Tratto il giorno Settembre 30, 2019 da https://docs.microsoft.com/en-us/cognitive-toolkit/

Nuscheler, D. (2016). *Regularly change a running system! An analysis of stage-specific criteria for attractingventure capital and changing the likelihood for getting funded.* International Finance and Banking Society.

Olimpio Guido, S. G. (2014, Dicembre 15). *Terrore a Sydney, ostaggi in un bar Blitz dopo oltre 16 ore: tre morti* . Tratto il giorno Ottobre 4, 2019 da Corriere.it: https://www.corriere.it/esteri/14_dicembre_15/tredici-ostaggi-un-bar-sydney-forse-attacco-islamico-04f48c1a-83ec-11e4-a2cc-02f7f9acc66f.shtml

pandas.read_csv - pandas 0.25.1 documentation. (2019, Agosto). Tratto il giorno Settembre 6, 2019 da pandas: powerful Python data analysis toolkit: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.read_csv.html

Post Intelligence - Buy Automatic Instagram Likes & View. (s.d.). Tratto il giorno Ottobre 2, 2019 da Post Intelligence: https://postintelligence.ai/

pydata.org. (2019, Luglio 18). *pandas: Python Data Analysis Library*. Tratto il giorno Agosto 13, 2019 da pydata.org: https://pandas.pydata.org/index.html

Python Software Foundation (US). (2019). *Welcome to Python.org*. Tratto il giorno 08 13, 2019 da python.org: https://www.python.org/

Python Software Foundation. (2019, Settembre 16). *datetime - Basic date and types - Python 3.4.7 documentation*. Tratto il giorno Settembre 16, 2019 da https://docs.python.org/3/library/datetime.html

Qzzr. (s.d.). *Create Online Quizzes That Drive Revenue* . Tratto il giorno Ottobre 2, 2019 da Qzzr.com: https://www.qzzr.com/

ResidentMario. (2019, Luglio). *missingno.* Tratto il giorno Settembre 10, 2019 da GitHub: https://github.com/ResidentMario/missingno

Rousseeuw, P. J. (1987). Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. *Computational and Applied Mathematics*(20), 53-65. doi:10.1016/0377-0427(87)90125-7

Russian Foundation for Advanced Research Projects in the Defense Industry. (2017, Dicembre 21). *FEDOR - The first Russian anthropomorphic robot] (in Russian)*. Tratto il giorno Settembre 30, 2019 da https://fpi.gov.ru/projects/khimiko-biologicheskie-i-meditsinskie-issledovaniya/fedor/

Shafranovich, Y. (2005, Ottobre). *Request for Comments: 4180.* Tratto il giorno Agosto 13, 2019 da IETF Tools: https://tools.ietf.org/html/rfc4180

SHALEV-SHWARTZ, S., & BEN-DAVID, S. (2014). *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press.

sklearn.org. (s.d.). *sklearn.model_selection.GridSearchCV*. Tratto il giorno Ottobre 4, 2019 da scikit-learn 0.21.3 documentation: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

STEIN, B. (2005). Fuzzy-fingerprints for text-based information retrieval. *Proceedings of the 5th international conference on knowledge management (I-KNOW 05), Graz, Journal of Universal Computer Science.*, 572-579.

Tata, A. D.-M. (2017). The psycholinguistics of entrepreneurship. *Journal of Business Venturing Insights, 7*, 38 - 44.

Tay Tweets. (s.d.). Tratto il giorno Ottobre 4, 2019 da Twitter: Twitter

TechCrunch – Startup and Technology News. (s.d.). Tratto il giorno Settembre 30, 2019 da TechCrunch: https://techcrunch.com/

Tesla Italia. (2019, Settembre). *Model S*. Tratto il giorno Settembre 30, 2019 da teslamotors: https://www.tesla.com/it_IT/models

Treccani.it. (2019, Settembre). *IPO (Initial Public Offering) in "Dizionario di Economia e Finanza"*. Tratto il giorno Settembre 10, 2019 da http://www.treccani.it/enciclopedia/ipo_%28Dizionario-di-Economia-e-Finanza%29

Uber Technologies inc. (s.d.). *Uber: guadagna denaro al volante o richiedi una corsa ora*. Tratto il giorno Settembre 15, 2019 da Uber: https://www.uber.com/it/it/

Vijayenthiran, V. (2018, Settembre 17). *Navya already sells fully self-driving cars, including in US.* Tratto il giorno Settembre 30, 2019 da Motor Autority: https://www.motorauthority.com/news/1118809_navya-already-sells-fully-self-driving-cars-including-in-us

Volkswagen Group Italia. (s.d.). *A8 L 2019 > Gamma Audi A8* . Tratto il giorno Settembre 30, 2019 da Audi Italia: https://www.audi.it/it/web/it/modelli/a8/a8-l.html

Wikipedia. (2019, Settembre). *Stock exchange.* Tratto il giorno Settembre 10, 2019 da Wikipedia, L'enciclopedia libera.