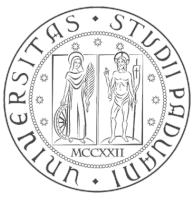


UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

TESI DI LAUREA MAGISTRALE

APPLICAZIONE DI TECNICHE DI MACHINE LEARNING PER L'ANALISI DEL RUOLO DEL CAPITALE UMANO NELLE STARTUP

Relatore: Prof. MORENO MUFFATTO

Candidato: FRANCESCO BRIGO

Matricola: 1179345

9 Dicembre 2019

ANNO ACCADEMICO 2018-2019

A tutti quelli che mi sono stati vicino durante questo percorso, alla mia famiglia per avermi aiutato e consigliato, a Stella per avermi sopportato, a hoLele per avermi accompagnato fino alla fine

Sommario

L'obiettivo di questo lavoro è quello di studiare l'apporto del capitale umano, inteso come l'insieme delle esperienze lavorative e formative, nel determinare la possibilità di successo di una startup. In particolare, in questo lavoro, verrà implementato un modello per la predizione automatica del successo delle startup a partire dalle caratteristiche del team di founder. Dai risultati ottenuti verrà poi estratto uno studio dell'importanza delle singole *feature* sulla predizione, delineando quindi le caratteristiche più importanti per un team di successo. In particolare è stata considerata di successo una startup che ha effettuato una exit, sia *IPO* che acquisizione, oppure che sia strutturata come azienda indipendente dopo almeno 12 anni dalla fondazione.

L'approccio seguito per realizzare questa ricerca può essere suddiviso in tre passi:

- creazione del dataset;
- feature engineering;
- implementazione dei modelli di classificazione.

Inizialmente l'interesse principale è stato quello di creare il dataset e ci si è concentrati sull'estrazione dei dati dalle varie fonti disponibili online. A partire da un database a disposizione di 18.000 nomi e 10.203 imprese si sono selezionati in maniera randomica 8731 soggetti e le relative 4753 startup.

Successivamente, nella fase di *feature engineering*, si sono elaborate queste informazioni per delineare un profilo del singolo soggetto: in dettaglio si sono analizzati parametri quali le esperienze lavorative pregresse, sia in termini di durata della carriera che di posizioni cambiate, la formazione accademica, sia in termini di titolo di studio conseguito che di ambito e di qualità dell'università che ha rilasciato tali titoli, ed infine l'eventuale fondazione di altre imprese da parte dei singoli soggetti. Infine, si è conclusa la fase di creazione del dataset delineando un profilo comune a tutti i founder della singola startup, considerando i dati relativi ai singoli membri del team e pesandoli in base ad una funzione *custom* che tiene conto della numerosità del team e delle conseguenti dinamiche di squadra.

L'ultimo passo è stata la creazione dei modelli di classificazione: sulla base dei dati a disposizione e della funzione di pesatura, si è scelto di utilizzare le moderne tecniche di *Machine Learning*. In particolare, sono stati proposti e valutati cinque modelli basati su regressione logistica, alberi di decisione, Random Forest, k-Nearest Neighbors ed una rete neurale.

Una volta determinato il modello migliore, che è risultato essere quello che utilizza le Random Forest, si è poi visualizzata l'importanza delle singole feature sul risultato finale, verificando le ipotesi di lavoro formulate a priori.

Gli sviluppi futuri ottenibili a partire da questo lavoro includono l'integrazione di ulteriori informazioni al dataset, si pensi ad esempio al settore della startup ed all'ambito di studio/lavoro dei founder, e il perfezionamento del riconoscimento delle feature già implementate utilizzando algoritmi di text mining o tecniche di term embeddings come *SkipGram* e *CBOW (Continuous Bag Of Words)*.

Indice

Capitolo 1

Il ruolo del capitale umano nel successo e nella valutazione delle startup

In questo capitolo verranno presentate le principali teorie sulla correlazione fra profili dei componenti del team di fondatori e/o management di una startup e il successo, che in questo lavoro, è stato inteso come la quotazione sul mercato azionario (*IPO*) o l'avvenuta acquisizione da parte di un'altra impresa o ancora come il rimanere strutturata come un'impresa indipendente dopo almeno 12 anni dalla fondazione.

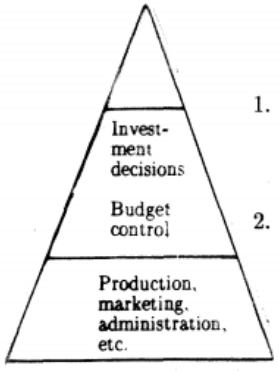
Fin dagli anni '80, gli studiosi si sono interessati alla correlazione fra competenze ed esperienze del *team* che guida l'azienda e il successo della stessa. Uno dei primi contributi a questo campo è stato quello proposto da Eliasson [2], in cui propone una visione dell'azienda come estensione del *team* manageriale. In particolare lo studio propone di considerare come fondamentali, in una prospettiva aziendale, tre concetti:

- **lo spazio di opportunità:** l'occasione di business nello specifico settore;
- **la conoscenza tacita:** la conoscenza difficilmente trasferibile, tipica dell'individuo e delle aziende, vista un vantaggio competitivo proprio per la difficoltà nell'esportarla [3];
- **la competizione:** il libero mercato che pone le aziende di fronte a costanti sfide tecnologiche ed economiche.

Punto focale della ricerca di Eliasson è che un team competente, a livello di management, è in grado di fornire i risultati migliori per tutte e tre queste aree. Infatti, il management è direttamente responsabile sia delle decisioni sul business e sugli sviluppi aziendali, che sull'assunzione di nuovo personale, controllando di fatto tutta l'azienda. Le decisioni aziendali, come in fig.1.1, possono essere divise in tre grandi categorie:

1. decisioni strategiche, ossia quelle che possono cambiare l'organizzazione della compagnia (ad esempio le assunzioni);
2. decisioni di controllo e coordinamento, ossia quelle relative alle attività di una singola *business unit*;
3. decisioni operative, relative al personale incaricato di testare e migliorare le *performance* delle singole attività.

Tutte queste tipologie di decisioni si basano sulla conoscenza tacita specifica dell'area, tuttavia la gerarchia delle decisioni, illustrata sempre in fig.1.1, mostra bene come le decisioni strategiche influenzino direttamente le altre due e, di conseguenza, l'intero flusso di decisioni aziendali dipenda da quelle del management.



	Type	Taken by and where	Answers question	Competence creation
1.	Strategic selection (affecting structures)	Top competent team CEO Board owners	What?	Selection Varied career Experience
2.	Control (Coordination)	CHQ Staff	In which order?	Education College, and University
	Operational (rationalization)	Locally	How?	Training Vocational

Source: Eliasson (1985, p. 14).

Figura 1.1: La piramide delle decisioni secondo Eliasson [2].

Risulta dunque ovvio che un management competente produca effetti positivi lungo tutta la filiera decisionale aziendale, incrementando, di conseguenza, il rendimento della società. Inoltre, poiché la conoscenza tacita aziendale è fondamentale nel lavoro quotidiano della società, coloro che aspirano a diventare parte del team manageriale devono risalire tutta la catena decisionale, partendo dal livello più basso. In definitiva, dunque ogni componente del team al livello più alto dipenderà dal team precedente, e così via, essendo ogni azienda un'entità dipendente dalla sua stessa memoria organizzativa. Inoltre, Eliasson delinea anche sei particolari caratteristiche che le varie competenze del management possono portare all'azienda:

1. intuizione, intesa come senso dell'orientamento, sia sul mercato che sulla situazione tecnologica;
2. propensione al rischio;
3. efficienza nell'analisi, intesa come l'abilità di identificare gli errori nel business;
4. efficacia nel correggere questi errori;
5. efficacia nel coordinare;
6. efficacia nel tradurre le esperienze all'interno dell'azienda.

Le conclusioni dell'autore evidenziano come esista una stretta correlazione fra le competenze dei lavoratori di una azienda e, sul lungo termine, il successo della stessa.

Nelle prossime pagine verranno presentate e discusse le principali teorie relative ad una correlazione fra successo (e quindi indirettamente valutazione) delle startup e profilo dei loro team manageriale o di founder.

Nel caso delle startup la riuscita (e quindi un possibile finanziamento esterno, con conseguente valutazione) è principalmente connessa, nei primi anni, al ruolo ed al lavoro che viene svolto dai founders stessi. Infatti, generalmente le attività di prototipazione e di definizione del modello di business vengono sviluppate interamente dai fondatori che, successivamente, le presentano ai possibili finanziatori. La composizione del team e il profilo accademico e lavorativo dei founder fornisce quindi un primo biglietto da visita per la startup, aumentando o diminuendo la possibilità di ottenere investimenti.

Un'analisi più accurata di questo fenomeno viene svolta in [4], in cui gli autori inseriscono, fra i 35 criteri di decisioni (in fig.1.2) dei *Venture Capitalist* europei, le competenze e le esperienze del management . L'ipotesi sviluppata ed analizzata da Myzuka et al. riguarda

la preponderanza di queste caratteristiche rispetto alle restanti. Per dimostrare questa intuizione gli autori hanno effettuato un'analisi statistica utilizzando tecniche di clustering proposte da Statistical Package for Social Science (SPSS), software di IBM. In dettaglio gli autori hanno raccolto interviste da oltre 70 *VC (Venture Capitalist)* europei e i risultati ottenuti vedono la formulazione di tre cluster:

- Investitori nazionali, un gruppo di 18 VC, in cui sono riconosciuti come elementi chiave le capacità di leadership dell'imprenditore e del suo team e, solamente successivamente si considerano le competenze del team stesso (soprattutto riguardo Marketing e Vendite, Processi e produzione). Inoltre, questi investitori si focalizzano su mercati *nazionali*, da cui il nome dato dagli autori, in quanto maggiormente interpretabili e più comprensibili, anche a livello di competizione;

- I dealer, costituiti da 4 investitori, sono l'unico gruppo in cui prevalgono gli indici finanziari (*time to breakeven, abilità a livello amministrativo/contabile del team* e riguardo l'accordo in sè (adeguatezza al precedente portfolio dell'investitore, caratteristiche dell'accordo stesso);

- gli investitori *mainstream* costituiscono il gruppo più numeroso, reputano chiavi le competenze (sia amministrative/contabili sia di marketing e vendite, sia di processi e produzione) del team di management, oltre che la comprensione del mercato e la possibilità di avere entrate.

L'ultimo gruppo,insieme all'analisi complessiva, dimostra come tutti e cinque i criteri legati alle capacità del team erano tra i sette più importanti indicatori presi in esame dai VC, con la capacità di leadership dell'imprenditore capofila e del management team di gestione al primo e al secondo posto. Lo studio di Myzuka et al. conferma l'idea che avere un buon team può portare al successo anche un prodotto mediocre, mentre un buon prodotto con un team mediocre ha meno probabilità di successo.

TABLE 5 Cluster Mean Scores

	Final Rank	Cluster 1 (n = 18)	Cluster 2 (n = 4)	Cluster 3 (n = 51)	Mean Rank	SD	SL[a]
Financial criteria							
Time to break even	12	23.2[b]	7.6*[c]	13.6	15.6	9.7	0.000
Time to payback	20	20.1	18.0	18.1	18.8	9.0	0.494
Expected rate of return	11	13.5	22.2	14.6	14.7	9.0	0.212
Ability to cash out	9	12.0	18.6	12.2	12.5	8.1	0.291
Product–Market criteria							
Degree market already established	19	21.2	15.6	13.9	18.6	8.9	0.320
Market size	29	23.4	17.4	24.1	23.6	7.9	0.258
Seasonality of product–market	33	27.3	15.5*	26.1	25.8	8.2	0.027
Sensitivity to ecnomic cycles	30	26.4	27.9	23.1	24.2	8.9	0.300
Market growth and attractiveness	18	21.8	15.5	17.5	18.5	9.3	0.192
Uniqueness of product and technology	17	19.3	15.1	17.9	18.4	7.7	0.617
National location of business	27	14.2*	10.6*	26.0	22.3	10.2	0.000
Degree of product–market understanding	10	13.0	22.7	14.0	14.2	7.8	0.071
Strategic–Competitive criteria							
Ease of market entry	24	19.5	29.4	21.3	21.3	9.5	0.171
Ability to create post-entry barriers	14	18.3	12.2	16.0	16.3	9.0	0.418
Sustained share competitive position	6	15.5	14.9	9.8	11.4	8.6	0.037
Nature and degree of competition	26	15.1*	25.4	24.0	21.9	7.9	0.000
Strength of suppliers and distributors	25	18.7	17.0*	23.1	21.7	8.3	0.079
Fund criteria							
Business meets fund constraints	15	16.1	16.0	17.1	16.8	10.3	0.924
Business and product fit with fund portfolio	28	21.5	14.6*	24.7	23.4	9.3	0.065
Ability of investors to influence nature of business	21	22.8	23.7	17.4	19.1	8.8	0.047
Location of business relative to the fund	35	22.7	11.2*	28.9	26.3	8.2	0.000
Management team criteria							
Leadership potential of management team	2	9.0	20.9	7.9	8.9	7.4	0.003
Leadership potential of lead entrepreneur	1	7.3	18.2	7.9	8.3	7.0	0.012
Recognized industry expertise in team	3	16.2	10.9	8.6	10.6	6.9	0.000
Track record of lead entrepreneur	4	14.5	13.2	9.4	10.9	8.7	0.085
Track record of management team	5	13.2	27.2	9.2	11.2	8.5	0.000
Management competence criteria							
Marketing/Sales capabilities of team	7	11.4	18.9	11.0	11.5	6.5	0.065
Process/Production capabilities of team	16	14.5	23.0	18.1	17.5	9.4	0.189
Organizational/Administrative capabilities of team	8	16.8	12.5	10.9	12.4	8.0	0.025
Financial/Accounting capabilities of team	13	18.6	13.6	15.1	15.9	8.7	0.304
Deal criteria							
Stage of investment required	23	15.7*	20.0	22.9	21.0	10.4	0.039
Number and nature of co-investors in deal	32	24.1	15.7*	26.4	25.3	8.3	0.036
Ability to syndicate deal	31	21.2	16.7*	26.1	24.4	8.1	0.010
Scale and chance of later funding rounds	34	29.8	17.5*	25.3	26.0	8.6	0.018
Importance of unclear assumptions	22	10.9*	30.1	23.2	20.6	10.3	0.000

Figura 1.2: I 35 criteri di valutazione per i VC secondo Myzuka et al. [4], corredati da ranking totale e suddiviso per cluster

Dunque i risultati di questo studio [4] confermano l'idea che maggiori possibilità di successo vengano assegnate, da parte degli investitori, alle start-up composte da team con conoscenze e competenze eterogenee e variegate, oltre che importanti.

Altre ricerche, fra cui quello di Clarysse [6], sostengono che i founder delle startup abbiano maggior successo nell'attrarre investimenti se sono *imprenditori seriali*, ossia se hanno già esperienze di startup alle spalle. In definitiva il profilo dei founder, sia a livello accademico che lavorativo, interessa ai possibili investitori, e di conseguenza fornisce un possibile indicatore sulla riuscita della startup.

Altri studi, come ad esempio [7], hanno cercato di controllare l'impatto del cosiddetto *capitale umano*, ossia l'insieme delle caratteristiche dei lavoratori di un'azienda, nel successo a breve termine (orizzonte a 3 anni) dell'azienda. In particolare Baptista et al.[7] analizzano la situazione portoghese tramite il dataset Quadros de Pessoal (*Registro del personale*), fornito dal Ministero del lavoro. Tale dataset contiene informazioni su tutti i lavoratori

dipendenti privati dell'economia portoghese e sui loro datori di lavoro. I dati interessanti per gli autori riguardano il singolo dipendente e sono ad esempio: sesso, età, istruzione, competenze, occupazione, impiego, anzianità. Le ipotesi che gli autori vogliono verificar sono le seguenti:

1. le startup fondate da soggetti precedentemente disoccupati hanno meno possibilità di sopravvivere;

2. anche nel caso di competenze eterogenee queste hanno un impatto meno positivo nelle startup fondate da disoccupati rispetto alle altre.

Il modello di analisi utilizzato prevede le variabili, mostrate in fig.1.3, come rappresentazione del singolo founder.

Variable	Definition	No. obs[a]	Mean	Std. dev.	Min	Max
Survival	Variable = 1 if both firm and founder remain active 3 years after founding, and 0 otherwise	169,918	0.76	0.43	0	1
Log size	Log of the number of employees of the firm at founding	169,918	1.23	0.73	0	6.46
Emp education	Proportion of employees that are university graduates at founding	165,354	0.016	0.11	0	1
Age below 30	Variable = 1 if founder is 29 years of age or younger at the time of founding, and 0 otherwise	161,582	0.20	0.40	0	1
Age 30–39	Variable = 1 if founder is between 30 and 39 years of age at the time of founding, and 0 otherwise	161,582	0.36	0.48	0	1
Age 40–49	Variable = 1 if founder is between 40 and 49 years of age at the time of founding, and 0 otherwise	161,582	0.27	0.44	0	1
Gender	Variable = 1 if founder is female, and 0 if it is male	169,918	0.28	0.45	0	1
University grad	Variable = 1 if founder is a university graduate at the time of founding, and 0 otherwise	167,838	0.08	0.28	0	1
High school grad	Variable = 1 if founder is a high school graduate at the time of founding, and 0 otherwise	167,838	0.17	0.38	0	1
Work exp	Number of years since founder entered the labor market	169,918	1.58	2.20	0	8
Spinout	Variable = 1 if founder was working in an incumbent of the startup's industry in the year before founding, and 0 otherwise	169,918	0.07	0.25	0	1
Industry exp	Number of years founder worked in the startup's industry prior to founding	169,918	1.31	1.98	0	8
Managerial exp	Number of years the founder worked in one of the top three hierarchical levels listed in the data set	169,918	0.52	1.28	0	8
Entrep exp	Variable = 1 if individual has founded at least one firm in the past, but was not a business owner at founding, and 0 otherwise	169,918	0.16	0.37	0	1
Portfolio	Variable = 1 if individual owned at least another business at founding, and 0 otherwise	167,838	0.06	0.38	0	1
Entrep team	Variable = 1 if firm is being founded by more than one individual, and 0 otherwise	169,918	0.51	0.50	0	1
Employment	Variable = 1 if founder was employed in the year before founding	169,918	0.24	0.43	0	1
Year dummies	Year dummies for cohort estimations: Variables = 1 for firms founded in 1995 and 0 otherwise; 1997 and 0 otherwise; 1999 and 0 otherwise; 2001 and 0 otherwise					
Industry dummies	Industry dummies at the three-digit level					

Figura 1.3: Le variabili definite da Baptista in [7]

A partire dal dataset in esame, gli autori decidono di concentrarsi sullo studio di aziende con un solo founder, facendo quindi coincidere la rappresentazione del singolo individuo con quella dell'azienda. Questa decisione deriva dall'ipotesi che aziende con più founder beneficeranno di background e competenze eterogenee. Baptista et al. hanno cercato quindi di esaminare il ruolo svolto dalle capacità pregresse e dall'esperienza dei fondatori in relazione ai primi anni di vita delle varie società. Il dataset viene ulteriormente suddiviso in base all'occupazione lavorativa precedente alla fondazione della startup del founder. In particolare, a sostegno di quanto espresso finora, gli autori notano che le caratteristiche imprenditoriali rivestono un ruolo particolarmente importante nei primi anni di una start-up, quando la missione e l'organizzazione dell'impresa sono in fase di costituzione e di assunzione di personale chiave.

I risultati ottenuti attraverso l'utilizzo di un modello di regressione logistica, che verrà sfruttato anche all'interno di questo lavoro, sono i seguenti:

- le startup fondate da imprenditori precedentemente occupati hanno una maggior probabilità di successo rispetto a quelle fondate da disoccupati, tuttavia questa correlazione risulta debolmente supportata dai dati a disposizione degli autori;
- gli imprenditori "occupati" risultano molto importanti le variabili che tengono conto dei livelli di istruzione, dell'esperienza lavorativa e non nel settore di appartenenza che aumentano di molto la probabilità di sopravvivenza, mentre per gli imprenditori "disoccupati" queste variabili non mostrano apprezzabili variazioni di *speranza di vita* per l'azienda.

In ogni caso anche questo studio sul capitale umano concorda con i precedenti [4] [6], in quanto mostra che le caratteristiche più importanti per i founder sono:

- i livelli di istruzione: tanto più alto è il livello di istruzione dei fondatori tanto più alta è la probabilità di sopravvivenza a tre anni;
- esperienza lavorativa;
- esperienza nel settore;
- esperienza manageriale.

In definitiva, Baptista et al. confermano l'idea che il ruolo del capitale umano sia di grande impatto nel processo di fondazione e finanziamento delle startup.

In [8] Ratzinger et al. si sono concentrati sulle startup digitali, ossia legate al settore tecnologico. In particolare hanno analizzato 4953 aziende, al fine di vedere l'impatto dei background accademici nei confronti della ricerca e dell'ottenimento dei fondi. La caratterizzazione dei founder utilizza le variabili mostrate in fig.1.4, che spaziano dal background accademico, suddiviso per campo, al sesso dei fondatori.

	Overall	Higher education		
		Technical	Business	General
Investment milestones				
Self-sustained	53.6%	45.4%	49.5%	46.8%
Funded	40.1%	45.3%	44.0%	45.9%
Exited	6.2%	9.4%	6.5%	7.3%
Higher education				
Technical	34.5%	–	37.2%	28.9%
Business	27.8%	30.0%	–	32.2%
General	7.5%	6.3%	8.6%	–
Graduate founders	55.8%	–	–	–
Undergraduate	44.1%	81.2%	80.7%	84.6%
Postgraduate	21.6%	43.7%	44.4%	33.5%
Doctorate	5.5%	13.5%	5.9%	17.6%
Start-up				
Average business age	5.94	6.25	5.45	5.70
Average number of cofounders	2.01	2.35	2.28	2.42
Founders				
All male	72.7%	69.4%	73.2%	66.5%
All female	4.1%	2.9%	3.6%	8.6%
Mixed gender	2.6%	3.6%	4.0%	5.4%
Unknown	20.6%	24.1%	19.3%	19.5%
Social capital				
Avg. cofounders attending same university	–	23.8%	23.9%	21.1%
Entrepreneurial experience	19.3%	22.6%	19.7%	21.9%
Self-sustained start-up	11.2%	11.5%	10.4%	11.1%
Funded start-up	4.6%	5.7%	4.9%	5.1%
Exited start-up	3.6%	5.4%	4.4%	5.7%
Observations	4953	1710	1379	370

Figura 1.4: Le variabili definite da Ratzinger in [8]

L'analisi empirica affronta dagli autori prevede l'utilizzo del modello predittivo espresso da

$$s_f = \beta_0 EDUCATION_f + \beta_1 Z_f + u_i$$

dove la startup $f - esima$ è rappresentata da:

- l'investimento s, ossia il risultato della previsione, che può essere di tre tipi:
  - autofinanziata, valore 0;
  - finanziata, valore 1;
  - *exit*, valore 2.
- EDUCATION, rappresenta il grado di educazione raggiunta, in particolare vengono considerati i tre livelli (laurea, laurea magistrale e dottorato) nei campi tecnico, economico ed altri;
- Z rappresenta invece i dati relativi alla startup (età della startup, numero di *cofounder* e sesso degli stessi);
- la variable u_i rappresenta, invece, l'eventuale frequentazione della stessa università da parte dei confondatori.

Investment milestone	Coefficient	z-statistic
Higher education		
Technical	0.23***	6.26
Business	0.09**	2.44
General	0.14**	2.19
Start-up		
Age	0.05***	12.70
Number of cofounders	0.10***	5.92
Gender		
Female	−0.01	−0.17
Mixed	0.11	1.11
Unknown	0.15***	3.71
Social Capital		
Cofounders attending same university	0.23**	2.57
Entrepreneurial experience		
Self-sustained start-up	−0.42***	−6.66
Funded start-up	0.15*	1.81
Exited start-up	0.36***	3.94
Number of obs	4953	
Wald chi^2	420.67	
Prob > chi^2	0.00	
Pseudo R^2	0.05	

Figura 1.5: I risultati proposti da Ratzinger in [8]: ***,**,* significativi al 10%,5%,1%

I risultati ottenuti da Ratzinger et al. in [8] sono:

1. i team con almeno un founder con un background tecnico hanno meno probabilità di rimanere in autofinanziamento ed hanno maggiori probabilità di riuscire ad ottenere i finanziamenti dall'esterno e di effettuare una *exit*, tuttavia questo impatto diminuisce con titoli di livello superiore;

2. i team con almeno un founder che ha un titolo di dottorato in *business administration* hanno meno probabilità di rimanere in autofinanziamento e hanno una maggiore probabilità di riuscire ad ottenere i finanziamenti da terzi, mentre il background universitario di tipo economico-finanziario non ha effetti significativi;

3. i team con almeno un founder che ha un background accademico di tipo umanistico hanno meno probabilità di rimanere in autofinanziamento e hanno una maggiore probabilità di riuscire ad ottenere i finanziamenti da terzi ed eventualmente una *exit*,

tuttavia l'impatto dei titoli umanistici non aumenta con l'aumentare della formazione (laurea magistrale e dottorato).

Altri studi, fra cui [9], riportano invece come l'eterogeneità all'interno del team sia fondamentale per la riuscita della startup. I dati utilizzati provengono dal dataset PSED II (*Panel Study of Entrepreneurial Dynamics*) [10], ossia una ricerca dell'Università del Michigan sul processo di formazione di imprese. In particolare il dataset utilizzato contiene i dati di oltre 1200 nuovi imprenditori, contattati a 12,24,35,48,60 e 72 mesi di distanza dall'intervista iniziale. Questo dataset è stato utilizzato principalmente per due ragioni: la prima riguarda la natura degli imprenditori intervista, nessuno di essi infatti ha già un'azienda avviata, la seconda è il tracciamento a 6 anni delle nuove aziende.

Nella ricerca effettuata da Bullon et al. [9] si analizza l'impatto del capitale umano sulla sopravvivenza della startup, in particolare vengono considerate le variabili *education*, in cui viene inserito il massimo grado di formazione raggiunto, *work-experience*, in cui vengono riportati gli anni di carriera lavorativa e *managerial experience*, in cui vengono riportati il numero di anni in cui i membri del team hanno avuto incarichi manageriali. Oltre a queste caratteristiche personali, gli autori hanno utilizzato un insieme di variabili rappresentanti, fra gli altri, l'eterogeneità del team, i finanziamenti e la grandezza del team. L'insieme della variabili, estratto dal dataset PSED II (Panel Study of Entrepreneurial Dynamics), è riassunto nella fig 1.6.

Table 1. Main descriptive statistics – teams with two or more members.

Variable	Mean	SD	Min	Max
New firm	0.296	0.457	0	1
Team-level variables				
Resource heterogeneity	4.261	1.327	1	6
Inequality in resource contributions	0.040	0.060	0	0.363
Team size	2.143	0.351	2	3
Previous industry experience	7.265	10.353	0	51
Previous startup experience	1.847	2.320	0	13
Male	0.589	0.493	0	1
Spousal team	0.523	0.500	0	1
Domestic commitments				
Married partner	0.718	0.451	0	1
No. of children	0.969	1.151	0	4
Social capital				
Self-employed parent	0.526	0.500	0	1
Relatives self-employed	0.341	0.475	0	1
Friends self-employed	0.401	0.491	0	1
Human capital variables				
Age	42.362	12.866	18	83
Work experience	20.460	12.039	0	64
Managerial experience	10.836	10.347	0	60
Education: up to high school	0.237	0.426	0	1
Education: college or vocational education	0.383	0.487	0	1
Education: college graduate	0.380	0.486	0	1
Financial capital				
Log (total household income)	10.964	0.744	7.824	13.682
Log (total funding)	0.304	3.842	−6.908	5.704
Competition				
Many	0.352	0.478	0	1
Few	0.491	0.501	0	1
No	0.157	0.021	0	1
High-tech startup	0.244	0.430	0	1
Log (organizing time)	6.230	1.035	0	9.320

Source: PSED II data set.
Note: Sample size = 287 observations.

Figura 1.6: Le variabili introdotte nello studio di Bullon et al. [9].

Le ipotesi che gli autori volevano verificare sono tre:

1. l'eterogeneità delle figure presenti nel team dei founder influenzano positivamente la probabilità che la startup diventi un'azienda profittevole;
2. eventuali precedenti esperienze di fondazione di startup influenzano in maniera positiva le probabilità di riuscita della startup, calcolate a partire dall'eterogeneità del gruppo di founder;
3. il ruolo delle precedenti esperienze lavorative incide in maniera tangibile sulla probabilità di riuscita della startup, calcolate a partire dall'eterogeneità del gruppo di founder.

Lo studio utilizza modelli basati sulla regressione per valutare le correlazioni fra le variabili precedentemente identificate. In particolare vengono utilizzati tre modelli differenti:

- il modello 1 è utilizzato per valutare quelle che gli autori definiscono come *variabili di controllo*, ossia caratteristiche fra cui la grandezza del team, il rapporto dei contributi espressi all'interno del team, il genere, il capitale finanziario e quello umano.
- il modello 2 viene utilizzato per testare l'ipotesi 1)
- il modello 3, invece, valuta l'interazione fra l'eterogeneità del team e le esperienze di startup e lavorative, testando quindi le ipotesi 2 e 3.

I risultati dei tre modelli, visibili in fig.1.7 1.8, dimostrano come l'ipotesi 1 sia corroborata dai dati in quanto ad esempio, nel caso di team composti da due persone, la presenza di background differenti aumenta di 1,26 volte la probabilità di creare imprese profittevoli. Invece, per quanto riguarda l'ipotesi 2 non risulta sostenuta da evidenze ottenute dai dati PSED II, infatti L'interazione fra le esperienze pregresse di startupper e l'eterogeneità del background del team risulta marginale. Infine, l'ipotesi 3 risulta sostenuta dai dati estrapolati dal PSED II in quanto l'interazione fra l'eterogeneità presente nei team e l'esperienza dei membri porta ad avere un incremento del 1,156 la probabilità di fondare una startup in grado di sopravvivere e diventare profittevole.

In conclusione, dai lavori qui presentati, si evince che il ruolo delle esperienza lavorative pregresse e della formazione accademica risulta essere particolarmente importante per comprendere a pieno i possibili andamenti dell'impresa, compresi gli eventuali finanziamenti. In dettaglio:

- gli studi di Eliasson [2] riportano il grande impatto delle scelte del management sull'efficacia e sull'efficienza delle decisioni e dei processi aziendali. Nel caso delle startup il management è spesso composto dagli stessi founder, dunque questi ultimi incidono direttamente sull'andamento generale della loro impresa;

- Muzyka et al. riportano i 35 criteri che gli investitori considerano nel momento in cui decidono di finanziare una startup, i criteri legati al team ed alla sua composizione risultano ai primi 5 posti, delineando quindi la primaria importanza del capitale umano nelle scelte degli investitori;

- Baptista et al., esaminando la situazione portoghese, mostra come l'esperienza lavorativa in generale e nello specifico settore siano fra le caratteristiche più importanti quando si analizzano le startup e le relazioni fra il loro successo e l'apporto del team;

- Ratzinger invece, esamina, oltre al titolo accademico, anche il campo di studi determinando l'importanza della correlazione fra queste due variabili in riferimento alla possibilità di ottenere finanziamenti ed, eventuale, di effettuare una *exit*;

- Bullon et al. mostrano come background differenti e esperienze lavorative pregresse influiscano sul successo della startup, mentre precedenti esperienze di fondazione di startup non incidono sullo stesso.

Table 3. Estimation results on the relationship between resource heterogeneity and the likelihood of profitable firm creation – teams with two or more members.

	Model 1			Model 2			Model 3		
	Coeff.	S.E.	Sig.	Coeff.	S.E.	Sig.	Coeff.	S.E.	Sig.
Constant	−4.981	3.037	*	−5.242	3.086	*	−4.126	3.141	
Resource heterogeneity	–	–		0.235	0.123	*	0.171	0.095	*
Inequality in resource contributions	–	–		3.687	2.377		3.688	2.411	
Previous startup experience	–	–		−0.077	0.074		−0.099	0.289	
Res. heterogeneity × previous startup exp.	–	–		–	–		0.008	0.062	
Previous industry experience	–	–		0.014	0.015		0.145	0.086	*
Res. heterogeneity × previous industry exp.	–	–		–	–		0.033	0.017	**
Team size	−0.662	0.457		−0.808	0.478	*	−0.854	0.495	*
Spousal team	−0.468	0.357		−0.468	0.371		−0.484	0.374	
Male	0.242	0.300		0.311	0.306		0.300	0.308	
Domestic commitments									
Married partner	−0.245	0.389		−0.154	0.406		−0.115	0.411	
No. of children	0.288	0.149	**	0.291	0.154	**	0.294	0.156	*
Social capital									
Self-employed parent	−0.067	0.293		−0.051	0.297		0.008	0.304	
Relatives self-employed	0.187	0.307		0.152	0.313		0.176	0.317	
Friends self-employed	0.120	0.292		0.139	0.298		0.127	0.303	
Human capital									
Age	−0.014	0.024		−0.018	0.025		−0.020	0.026	
Work experience	−0.010	0.026		−0.005	0.027		−0.003	0.028	
Managerial experience	0.037	0.021	*	0.033	0.023		0.032	0.023	
Education: up to high school	−0.337	0.418		−0.269	0.425		−0.309	0.430	
Education: college /vocational	0.358	0.341		0.367	0.349		0.341	0.352	
Education: college graduate	–	–		–	–		–	–	

(*Continued*)

Figura 1.7: I modelli valutati da Bullon et al. [9].

Table 3 – *continued*

	Model 1			Model 2			Model 3		
	Coeff.	S.E.	Sig.	Coeff.	S.E.	Sig.	Coeff.	S.E.	Sig.
Financial capital									
Log (total household income)	0.418	0.231	*	0.394	0.233	*	0.366	0.237	
Log (total funding)	0.136	0.043	***	0.133	0.044	***	0.126	0.045	***
Competition									
Many	−0.145	0.448		−0.337	0.477		−0.344	0.480	
Few	0.002	0.424		−0.102	0.442		−0.062	0.448	
No	–	–		–	–		–	–	
High-tech industry	−0.290	0.350		−0.287	0.358		−0.290	0.362	
Log (organizing time)	0.185	0.149		0.146	0.154		0.154	0.155	
Wald test	$\chi^2(19) = 35.28$			$\chi^2(23) = 43.78$			$\chi^2(25) = 48.63$		
	Prob $\geq \chi^2 = 0.0000$			Prob $\geq \chi^2 = 0.0031$			Prob $\geq \chi^2 = 0.0031$		
Number of observations	287			287			287		

Source: PSED II data set.
Notes: S.E., standard error.
*$p < 0.10$; **$p < 0.05$; ***$p < 0.01$.

Figura 1.8: I modelli valutati da Bullon et al. [9].

Capitolo 2

Obiettivo e metodologie

2.1 Obiettivo

Come mostrato nel capitolo 1 l'apporto del capitale umano nella fase di ricerca ed ottenimento degli investimenti risulta di fondamentale importanza. L'obiettivo di questo lavoro si pone quindi nel solco delle ricerche già presentate al fine di creare un modello di classificazione per valutare, a partire dalle caratteristiche del team di founder, il possibile successo della startup.

In particolare il presente lavoro sarà diviso in quattro fasi principali:

1. individuazione del problema, inteso come riconoscimento del task di apprendimento automatico da utilizzare e delle relative metodologie;
2. creazione del dataset e modellizzazione dei dati, che sarà illustrata nel cap.**??**;
3. creazione ed esecuzione dei modelli di apprendimento automatico, illustrata nel cap.4;
4. valutazione dei modelli e verifica delle ipotesi di lavoro, illustrata nel cap.5.

La problematica affrontata in questo lavoro è riassumibile come l'analisi delle caratteristiche del team di founder di startup e lo sviluppo di un modello di intelligenza artificiale per la predizione del successo dell'impresa a partire dai dati delle caratteristiche dei suoi founder. Per raggiungere l'obiettivo proposto in questo lavoro si è scelto di approcciare la problematica come un problema di classificazione binaria, in cui le imprese possono avere successo o meno. In questo capitolo verranno sintenticamente presentati i metodi scelti per affrontare la classificazione e la cui implementazione verrà discussa nel cap.4.

2.2 Regressione Logistica

In questo paragrafo si presenteranno i concetti base relativi alla regressione logistica [12] [13].

La regressione è una tecnica di analisi dei dati il cui scopo è la previsione e modellizzazione di una variabile dipendente sulla base di variabili esplicative già date. I modelli di regressione disponibili sono principalmente due: logistica, che verrà utilizzata in questo lavoro, e lineare. La regressione logistica differisce da quella lineare per la natura della variabile dipendente, essa infatti deve essere dicotomica.

La regressione logistica, parte dei modelli lineari generalizzati, studia la correlazione fra questa variabile dipendente e le variabili indipendenti attraverso una funzione non lineare. Il primo passo per lo studio della classificazione di variabili binarie è la riscrittura delle probabilità di successo ed insuccesso relative alla caratteristica x come probabilità condizionata:

$$\begin{aligned} P[Y = 1|x] &= p(x) \\ P[Y = 0|x] &= 1 - p(x) \end{aligned} \tag{2.2.1}$$

Dunque la variabile di output Y, ossia la classe di appartenenza dell'istanza, è modellabile come una variabile aleatoria di Bernoulli, di parametro p e di media μ. Inoltre, Y sarà la risposta del predittore lineare, ossia la quantità definita dal primo addendo in eq.2.2.2. Essa sarà calcolata in base ad una certa funzione g, tale per cui la risposta diventi limitata

fra 0 e 1. La funzione che realizza questa limitazione è la *mean function*, ossia una funzione non lineare definita tra 0 e 1, invertibile in cui l'inversa è la *link function*.

$$g(p(x)) = \beta \cdot x + b \tag{2.2.2}$$

La *mean function* utilizzata nella regressione logistica, e una delle più utilizzate, è la funzione logit che è definita come segue:

$$logit(p) = log\frac{p}{1-p} = \beta \cdot x \tag{2.2.3}$$

Data p la probabilità di successo, la funzione $\frac{p}{1-p}$ misura il numero di volte in cui il successo è maggiore dell'insuccesso e viene detta *odds-ratio*, dunque la funzione logit rappresenta il logaritmo della probabilità che un dato evento accada rispetto al fatto che non accada.

Il valore atteso di x è dunque calcolabile come:

$$E[Y|x] = p(x) = \frac{e^{\beta \cdot x}}{1 + e^{\beta \cdot x}} \tag{2.2.4}$$

ed è chiamato funzione logistica.

Per stimare il parametro β (o i parametri nel caso di più feature) si può utilizzare il principio della massima verosomiglianza (*maximum likelihood*), debitamente modificato come segue:

$$\mathcal{L}(\beta) = \prod_{i=1}^{n} f(y_i|x_i) = \prod_{i=1}^{n} p^{y_i}(x_i)(1 - p^{y_i}(x_i)) \tag{2.2.5}$$

Da cui deriva la funzione di verosimiglianza logaritmica

$$log\mathcal{L}(\beta) = \sum_{i=1}^{n} y_i(\beta \cdot x_i) - log(1 + e^{\beta \cdot x_i}) \tag{2.2.6}$$

Il problema della stima del parametro β si riduce, quindi, alla massimizzazione della funzione 2.2.6.

2.3 Decision Tree

Un'ulteriore tecnica di classificazione è chiamata *Decision Tree* [23] e prevede l'utilizzo di una struttura gerarchica ad albero per inserire l'insieme degli attributi da controllare al fine di predire la variabile di output. In maniera formale, come definito in [23], un albero di decisione è un classificatore che ricorsivamente partiziona lo spazio delle istanze. La struttura ad albero prevede quindi:

- una radice, ossia il nodo di partenza senza archi entranti;
- i nodi interni, a cui corrisponde un test su una o più variabili ed il cui risultato suddivide lo spazio dell'istanza in due o più sottospazi;
- archi, identificati dal valore delle variabili del nodo padre a cui sono collegati;
- foglie, ognuna delle quali rappresenta il valore predetto a cui si giunge seguendo il percorso, costituito dagli archi, dalla radice dell'albero alla foglia in analisi.

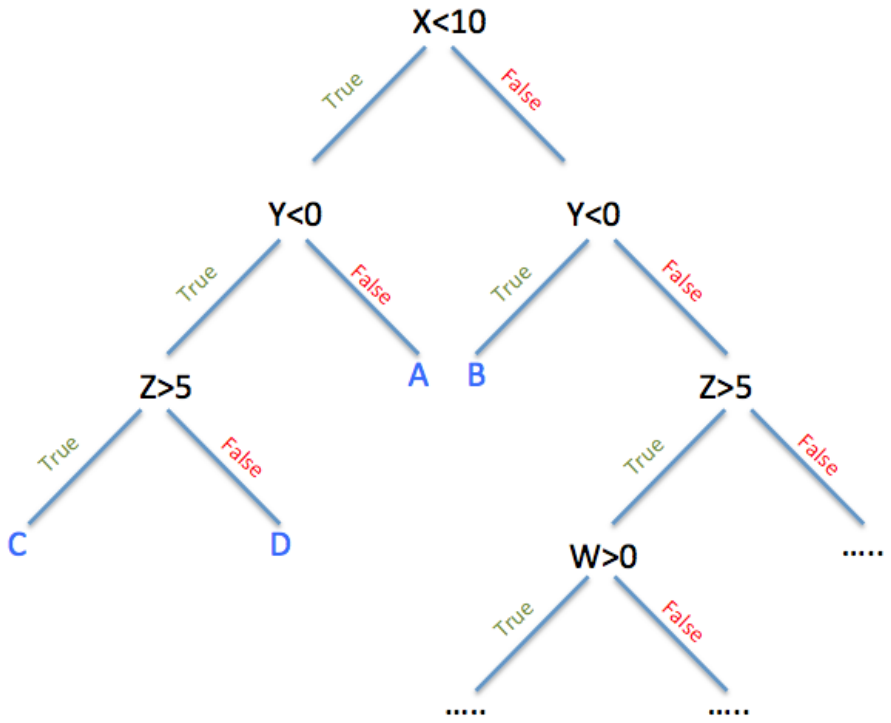


Figura 2.1: Un esempio di albero di decisione [23].

2.3.1 Apprendimento

L'apprendimento automatico che realizzala struttura ad albero dei decision tree è di tipo supervisionato, ossia consiste nel riuscire ad estrapolare le caratteristiche dell'albero a partire da un insieme di dati costituiti dalle variabili di input e dal relativo output. L'insieme dei dati utilizzati per allenare il decision tree viene detto *training set*.

Fra le principali caratteristiche dell'algoritmo di apprendimento c'è quella di trovare l'albero più piccolo, fra quelli validi, consistente con il training set. La soluzione di questo problema può prevedere più approcci: uno è quello di generare tutti gli alberi possibili e scegliere, solo successivamente, il più piccolo. Tuttavia, questo approccio farebbe ricadere la costruzione dei decision tree nella categoria dei problemi NP-C, ossia nella classe più complessa dei problemi non computabili in tempo polinomiale. Dunque, l'approccio più utilizzato per la creazione degli alberi decisionali è quello euristico: per ogni nodo viene scelto l'attributo (o gli attributi) che impattano maggiormente sulla classificazione, in tal modo la suddivisione può avere due risultati:

- determinazione della classe degli esempi, se questi ultimi hanno tutti lo stesso risultato nel test identificato dal nodo;
- ricorsione su un'altra feature, nel caso opposto.

La scelta maggiormente impattante è quella che riduce al minimo la varianza in ogni "strato" dell'albero. Ad ogni applicazione della regola *greedy* si possono dunque verificare due casistiche:

1. gli esempi con il dato valore per quella *feature* hanno *label* differenti bisognerà proseguire con la classificazione;
2. in caso contrario si può determinare la classe corretta per i dati a disposizione.

Il problema principale legato all'allenamento degli alberi decisionale è il cosiddetto ***overfitting***. Questa possibilità dipende, tipicamente, da un dataset con molti più attributi del necessario e/o con pochi esempi nel test set e, nel caso degli alberi di decisione, può portare ad alberi più estesi del necessario ed a classificazioni errate dovute a relazioni inesistenti fra feature e classe.

L'errore opposto all'overfitting è l'*underfitting* in cui l'albero è troppo ridotto e tralascia relazioni valide fra le feature e le classi, portando a classificazioni errate. Alcune tecniche che rimuovono il problema dell'overfitting ed, in generale, migliorano le prestazioni dell'apprendimento per i decision tree sono il pruning e la cross-validation che verranno presentate nei prossimi paragrafi.

Pruning

Il *pruning* (potatura) consiste nella riduzione della dimensione degli alberi per eliminare le scelte causate dall'overfitting. Generalmente gli algoritmi di pruning si dividono in:

- Pre-pruning: la potatura dei nodi viene effettuata contestualmente alla creazione dell'albero;
- Post-pruning: la potatura viene effettuata su un albero già creato, verificando per ogni nodo delle condizioni contenute in un insieme di dati definito *validation set*.

Cross Validation

La cross validation è una tecnica che prevede l'estrazione dal training set una parte dei dati che vengono utilizzati per verificare la predizione di eventi sconosciuti. I risultati della verifica andranno poi a determinare l'albero finale modificandone la struttura.

2.4 Random Forest

Il metodo delle foreste casuali (*Random Forest*) [14] è uno dei metodi d'apprendimento d'insieme (*ensemble learning*) ed è sostanzialmente costituito da un insieme di alberi di decisione. Infatti, i metodi di ensemble learning si basano sull'idea di combinare i giudizi di più classificatori in modo da migliorare la capacità di predizione effettuando una votazione, eventualmente pesata, dei singoli giudizi. In tal modo, nonostante un aumento della complessità computazionale, si ottengono classificatori più precisi e con un minor rischio di overfitting. Una delle possibili tecniche è il *bagging* (*BootStrap AGGregatING*) che, volendo ridurre i problemi relativi all'overfitting, addestra diversi classificatori su sottoparti del training set e, alla fine, esegue una votazione per maggioranza per classificare il singolo elemento. *Random Forest* appartiene agli algoritmi di bagging su Decision Tree. Gli alberi vengono creati similmente a quanto accade nei Decision Tree ma solo un sottoinsieme dei dati a disposizione per il training e delle feature da analizzare vengono inserite in ciascun albero. Questo meccanismo viene implementato modificando l'algoritmo in modo tale da permettere al *runtime* una scelta casuale di un sottoinsieme di feature, fra le quali verrà selezionata la migliore. In tal modo si riduce la qualità complessiva del singolo albero ma, complice la diversità fra gli alberi e il loro numero, si migliorano le prestazioni del classificatore.

La classe del singolo record sarà data quindi dalla scelta della maggioranza degli alberi presenti nella foresta, come illustrato in fig. 2.2

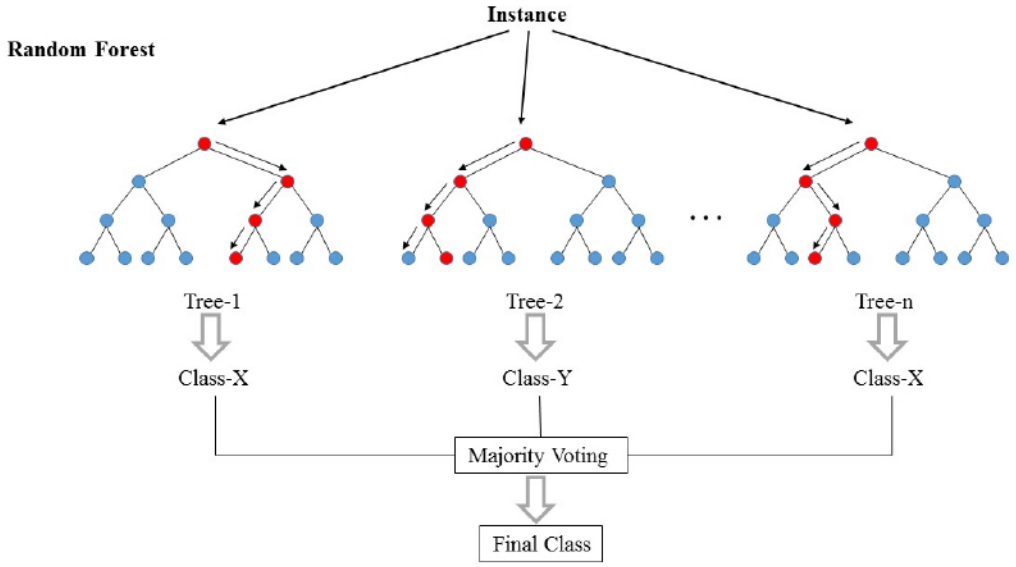


Figura 2.2: Un esempio di foresta casuale [14].

2.5 K-Nearest Neighbors

Questo metodo di classificazione appartiene alla classe dei classificatori supervisionati e si basa sul concetto di distanza.

2.5.1 Il concetto di distanza

Il concetto di distanza [15] è fondamentale per utilizzare in modo corretto ed efficiente l'algoritmo *K-Nearest Neighbors*. Infatti i punti, che rappresentano i singoli elementi dell'insieme da classificare, sono inseriti all'interno di uno spazio ad M dimensioni, dove M è il numero di feature degli elementi dell'insieme.

Tuttavia, esistono molteplici definizioni di distanza, fra cui:

1. distanza euclidea: dati due vettori $x = (x_1, ...x_n)$ e $y = (y_1, .., y_n)$ la distanza viene definita come $d(x, Y) = (\sum_{i=1}^{n} |x_i - y_i|)^{1/r})$;

2. distanza di Jaccard: dati due punti s e t la distanza viene definita come $d(s, t) = 1 - \frac{|s \cap t|}{|s \cup t|}$;

3. *cosine similarity*: per cui dati due vettori in n dimensioni la distanza viene calcolata come segue $d(x, y) = arccos(\frac{xy}{||x|| \cdot ||y||})$. La normalizzazione di questa distanza è da effettuare dividendo il risultato per $\pi/2$;

4. distanza di Hamming: calcolata semplicemente come il numero di coordinate in cui due vettori differiscono;

La scelta della funzione distanza può dipendere anche dalla natura delle *feature* che si analizzano(numeriche, logiche o stringhe) e può avere effetti rilevanti sulle performance dell'algoritmo [16].

2.5.2 Algoritmo KNN

L'algoritmo alla base del k-Nearest Neighbors [17] prevede una fase di apprendimento e una di test: durante la fase di apprendimento l'insieme di input I viene distribuito in uno spazio multidimensionale di dimensione N dove N è il numero di feature di ogni elemento

di I, inoltre ogni elemento di I è classificato correttamente a priori. Una volta terminata la fase di apprendimento vengono analizzati gli elementi dell'insieme da classificare e si dispongono anch'essi nel piano precedentemente identificato. L'algoritmo calcola quindi i k elementi più vicini per ognuno degli elementi dell'insieme da classificare e ne sceglie la relativa classe in base ad una votazione a maggioranza fra i vicini selezionati.

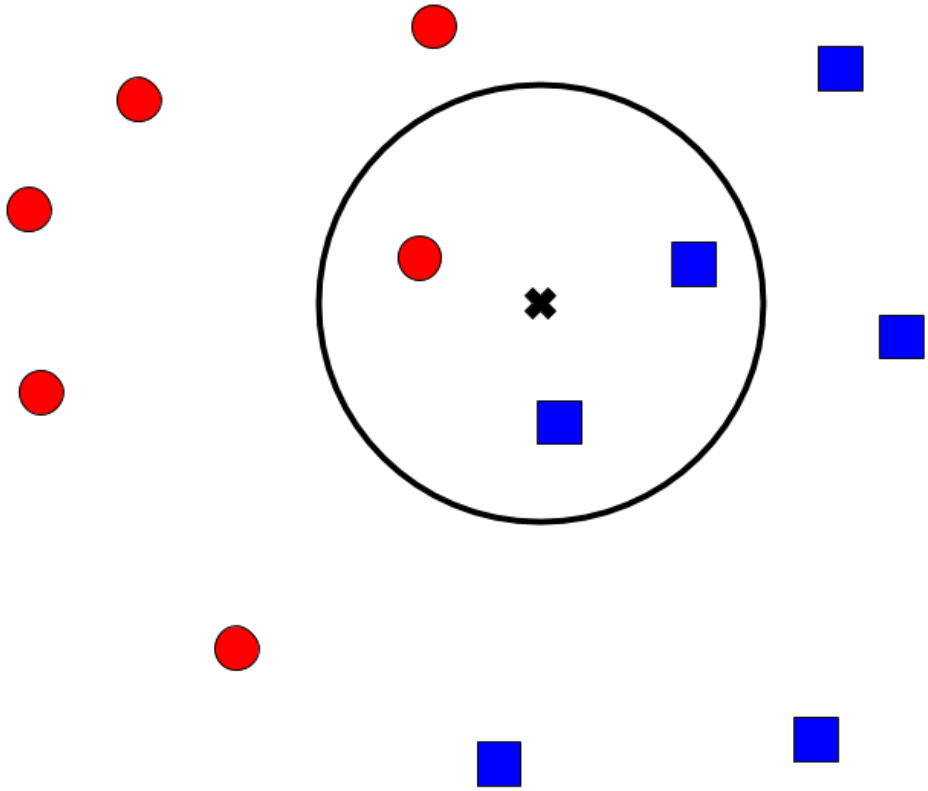


Figura 2.3: Un esempio di K-nearest neighbors, con k=3.

k-Nearest Neighbor
Classify (**X**, **Y**, *x*) // **X**: training data, **Y**: class labels of **X**, *x*: unknown sample
for $i = 1$ **to** m **do**
  Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set I containing indices for the k smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i$ where $i \in I\}$

Figura 2.4: L'algoritmo K- nearest neighbors [18].

Fra i principali vantaggi dell'algoritmo c'è la flessibilità più ampia nella classificazione grazie al concetto di distanza ed ai "contorni" delle singole classi. Tuttavia, l'utilizzo di un concetto non univoco di distanza e la sensibilità al rumore all'interno dell'algoritmo sono i principali svantaggi. Altro svantaggio è la richiesta di risorse computazionali e di memoria elevate, specie per insieme di input e di test numerosi.

2.6 Reti neurali

In questa sezione verranno presentati i concetti basilari e principali relativi alle reti neurali. Ci si focalizzerà sulle loro principali caratteristiche, in modo tale da avere un quadro generale introduttivo dei concetti che risultano comunque necessari per l'implementazione che è stata effettuata e verrà esposta in seguito.

2.6.1 Background storico

Il concetto di rete neurale venne introdotto da McCulloch e Pitts in *A logical calculus of the ideas immanent in nervous activity* per la prima volta nel 1943. In questo lavoro venne presentata una prima bozza di rete neurale tramite un combinatore lineare, con soglia di ingresso, i cui input erano dati binari e il cui output era un singolo dato binario. Combinando più dispositivi di questo tipo si potevano costruire tutte le funzioni booleane di base.

La scoperta che ha portato all'attuale modello di rete neurale è da attribuire a Rosenblatt [19] che, nel 1958, pubblica *Psychological review* dove teorizza il cosiddetto modello a percettrone, illustrato in fig.2.5, utilizzato per problemi di riconoscimento e classificazione di forme.

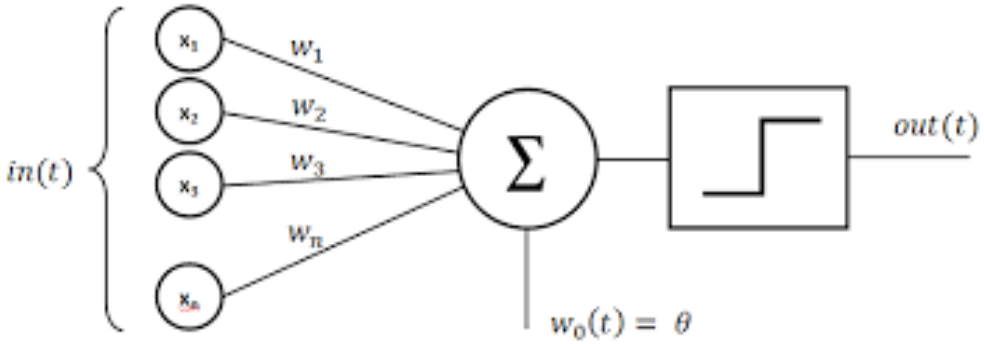


Figura 2.5: Il modello a percettrone di Rosenblatt [19].

L'altra grande scoperta è l'algoritmo di apprendimento e retropropagazione dell'errore (*error backpropagation*) proposto da Rumelhart, Hilton e Williams nel 1986 [**?**]. Questo algoritmo permette l'allenamento delle reti, superando le limitazioni del modello di Rosenblatt: ad esempio l'impossibilità di risolvere i problemi XOR.

Tuttavia, negli anni seguenti si è assistito ad un progressivo sviluppo di modelli alternativi come la support vector machine che hanno man mano soppiantato i modelli di reti neurali. Solamente a partire dalla fine degli anni 2000 si è riscoperta la teoria delle rete neurali ed ora sono largamente utilizzate in gran parte dei problemi scientifici come, ad esempio, l'analisi dei dati biologici, la computer vision e le analisi finanziarie.

2.6.2 Concetti base

Le reti neurali artificiali (*Artificial Neural Network, ANN*) sono modelli matematici ispirati nella struttura e nel funzionamento alle reti neurali biologiche. Per tanto risulta utile introdurre i concetti base delle reti neurali biologiche, partendo dalla struttura stessa del sistema nervoso umano.

Il sistema nervoso umano si basa su 10^{11} neuroni e 10^{15} connessioni ed affida alla variabilità delle connessioni e alla plasticità dei neuroni il funzionamento del cervello. Le relazioni che abbiamo con il mondo e l'apprendimento che ne consegue risultano alla base

del processo di variazione chimica che porta ad una variazioni sinaptica. DA CAMBIARE I cambiamenti appena descritti risultano ancoraQuesti cambiamenti sono alla base della progettazione delle reti neurali: infatti le reti neurali artificiali vengono generalmente presentate come dei sistemi di "neuroni" fra loro interconnessi, tra i quali avviene uno scambio di messaggi. Ciascuna connessione ha un relativo peso associato, detto *weight*; il cui valore è regolabile in base all'esperienza in esame e ciò rende le reti neurali uno strumento adattabile ai vari tipi di input e con la capacità di apprendere.

Il neurone artificiale

Il concetto chiave nel campo delle reti neurali è il neurone, illustrato schematicamente in fig. 2.6, che può essere visto come una unità computazione che prende come input x_1, x_2, x_3 e produce come risultato un y, variabile che viene detta attivazione del neurone.

Entrando più nello specifico, un neurone è un processo computazionale che, ricevuto un insieme di input I, può attivarsi, elaborando gli input per calcolare una sua specifica funzione, oppure rimanere inattivo. Il neurone sarà quindi caratterizzato da un valore di soglia di attivazione e da una funzione di attivazione. La funzione di attivazione determina l'output del neurone sulla base degli input che gli sono stati forniti secondo la formula seguente:

$$Y = f(\sum_{i=1}^{n} w_i X_i)$$

dove f è la funzione di attivazione, w_i il peso relativo all'i-esimo input e X_i l'i-esimo input. Generalmente, la scelta della specifica funzione di attivazione dipende dall'output che si vuole ottenere del singolo neurone, tuttavia le più comuni sono:

- la funzione sigmoide;
$$f(x) = \frac{1}{1 + e^{-x}}$$
- la funzione gradino;
$$f(x) = \begin{cases} 0 & \text{se } x \leq a \\ 1 & \text{se } x > a \end{cases}$$
- la funzione tangente iperbolica;
$$f(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} = tanh(x)$$

La scelta della funzione di attivazione permette anche, se scelta correttamente, di semplificare l'algoritmo di apprendimento da parte della rete.

Il processo di apprendimento

Nelle reti neurali il processo di apprendimento, ossia il metodo con cui si addestra la rete di neuroni, è riassumibile in tre paradigmi:

1. apprendimento supervisionato;
2. apprendimento non supervisionato;
3. apprendimento per rinforzo.

L'apprendimento supervisionato, illustrato in fig.2.7 ,prevede l'utilizzo di un insieme di esempi, detto *training set*, composto da coppie (x_i, y_{di}) dove:

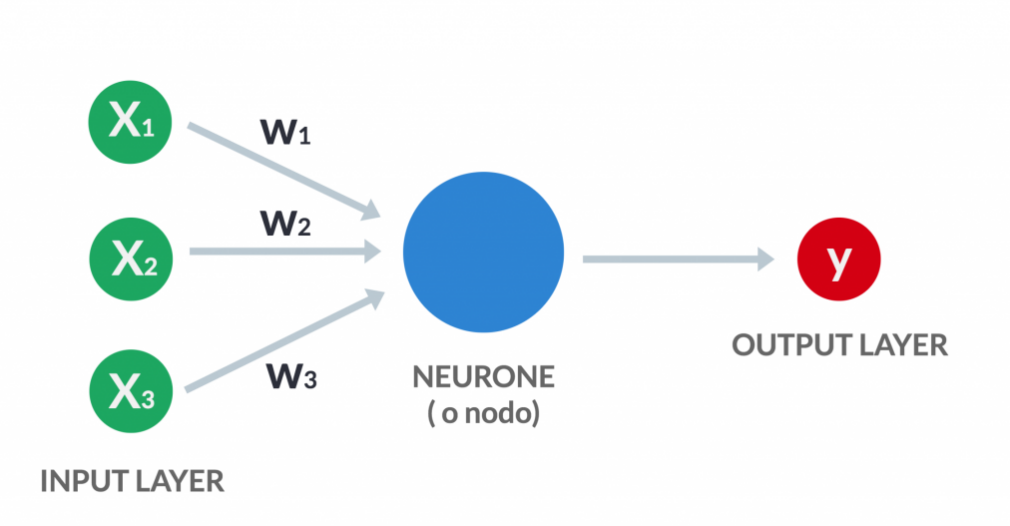


Figura 2.6: Il neurone artificiale.

- x_i rappresenta l'i-esimo input;
- y_{di} rappresenta l'i-esimo output impostato dall'utente.

Questa impostazione permette alla rete di adattare la legge di apprendimento per far risultare l'effettivo output y_i e y_{di} il più simile possibile. Le reti più comuni che utilizzano questo meccanismo di apprendimento sonom quelle multistrato.

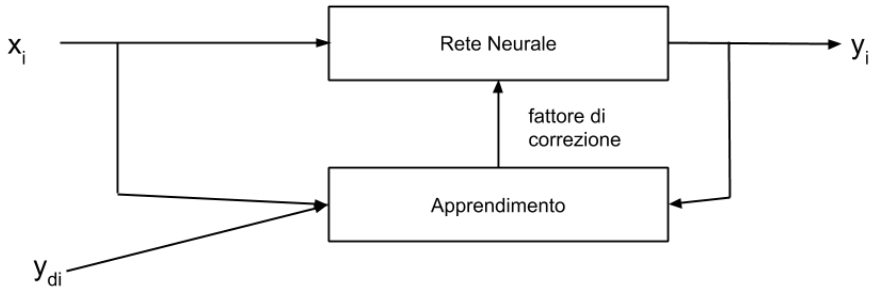


Figura 2.7: Il processo di apprendimento supervisionato.

Invece, l'apprendimento non supervisionato, presentato in fig.2.8, prevede che la rete analizzi e aggiusti i pesi in maniera totalmente autonoma, senza alcuna indicazione sull'output.

Infine, l'apprendimento per rinforzo prevede l'utilizzo di sensoristica, tramite i quali la rete neurale ha l'opportunità di ottenere dati sull'ambiente circostante che le permettono di determinare quale azione compiere. Poi, sulla base del risultato ottenuto l'agente, ossia la rete, può essere premiata o meno e, di conseguenza, adatterà il suo comportamento ai risultati ottenuti.

Tipologie di reti neurali

A seconda della caratteristica analizzata esistono diverso tipologie di rete neurali. Considerando le connesioni fra i vari neuroni si possono notare due tipi di rete:

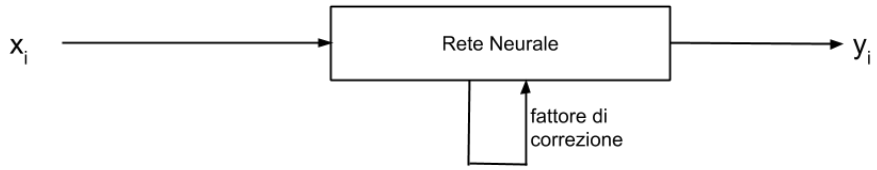


Figura 2.8: Il processo di apprendimento non supervisionato.

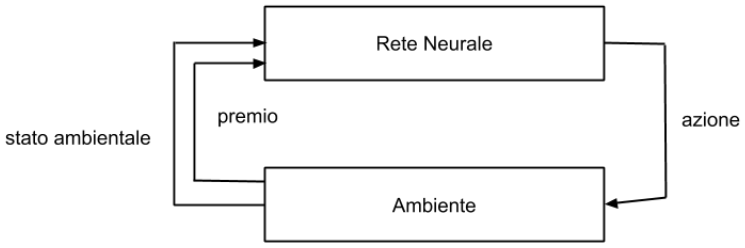


Figura 2.9: Il funzionamento dell'apprendimento per rinforzo.

- reti neurali *feedforwarding*;
- reti neurali *feedback*.

Le reti neurali *feedforwarding*, illustrate in fig.2.10, sono caratterizzate dall'assenza di cicli all'interno delle connessioni stabilite dai neuroni. Pertanto, all'interno della rete le informazioni viaggiano in un solo senso. Invece, le reti neurali *feedback* utilizzano connessioni fra i neuroni che formano un ciclo diretto, ossia con una singola direzione, creando quindi uno strato interno alla rete stessa. Questa caratteristica dona alla rete la possibilità di avere un comportante che varia dinamicamente nel corso del tempo. Inoltre, le reti *feedback* possono sfruttare una memoria interna per processare i diversi input.

Altre due tipologie di reti originano sempre dall'osservazione delle connessioni, in particolare esistono reti completamente connesse (*fully connected*) in cui ogni neurone è connesso con tutti gli altri, e reti stratificate, come quella visibile in fig.2.10, in cui i neuroni sono organizzati in strati, detti *layer*. La struttura di una rete *fully connected* prevede che le uniche connessioni esistenti siano quelle fra i neuroni appartenenti ad un layer e i neuroni dello strato successivo, procedendo verso l'uscita. Nel caso di una rete *fully connected*, si definiscono i seguenti layer fondamentali:

- **input layer**, ossia il gruppo di neuroni direttamente collegato con l'input;
- **output layer**, cioè il layer collegato all'uscita della rete;
- **hidden layer**, è lo strato (o gli strati) non direttamente collegato nè con gli input nè con l'output.

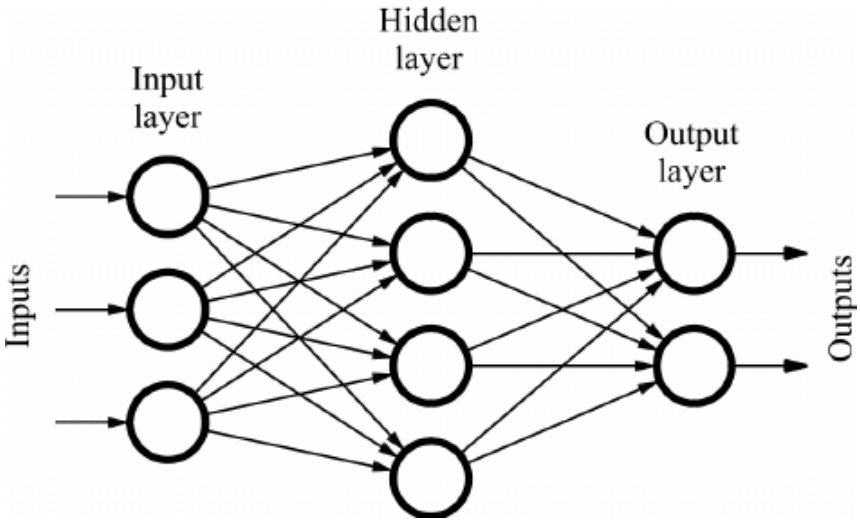


Figura 2.10: Un esempio di rete feedforwarding [20].

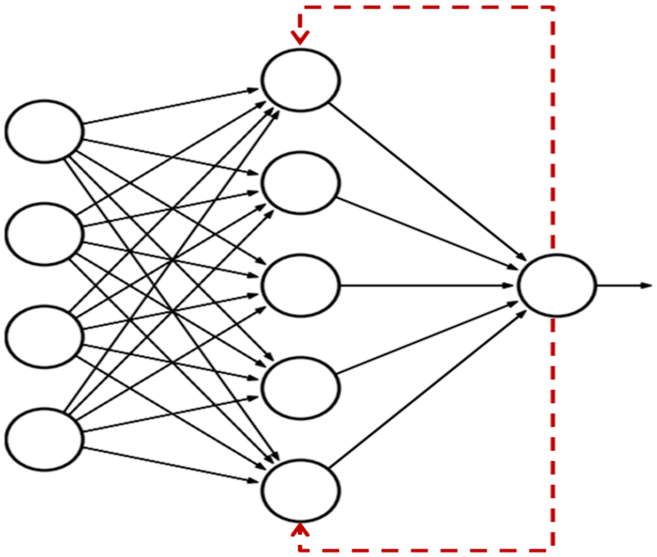


Figura 2.11: Un esempio di rete feedback [21].

Allenamento con backpropagation

In particolare l'algoritmo di *backpropagation*può essere suddiviso in due fasi: ***forward propagation*** e ***backward propagation***. Nella prima fase l'algoritmo determina tutte le attivazioni dei vari neuroni della rete usando dei pesi preimpostati, mentre nella seconda fase si controlla l'output della rete e la si confronta con l'uscita desiderata, calcolando quello che di fatto è l'errore fra le due. Successivamente si procede con la propagazione dell'errore calcolato in direzione opposta a quelle delle varie connessioni fra neuroni. Al termine si otteranno i pesi modificati in maniera tale da minimizzare l'errore precedentemente calcolato. L'intero algoritmo viene poi iterato più volte al fine di ottimizzare i pesi calcolati.

La base su cui si poggia l'intero algoritmo di backpropagation è la cosiddetta *Stochastic Gradient Descent (SGD)*. Quest'ultimo è un algoritmo utilizzato nell'ottimizzazione di funzioni differenziabili, in particolare utilizza la formula 2.6.1

$$w^{(t)} = w^{(t)} - \eta \nabla L_S(w^{(t)}) \tag{2.6.1}$$

dove L_S è la funzione che rappresenta l'errore considerando solamente le coppie che fungono

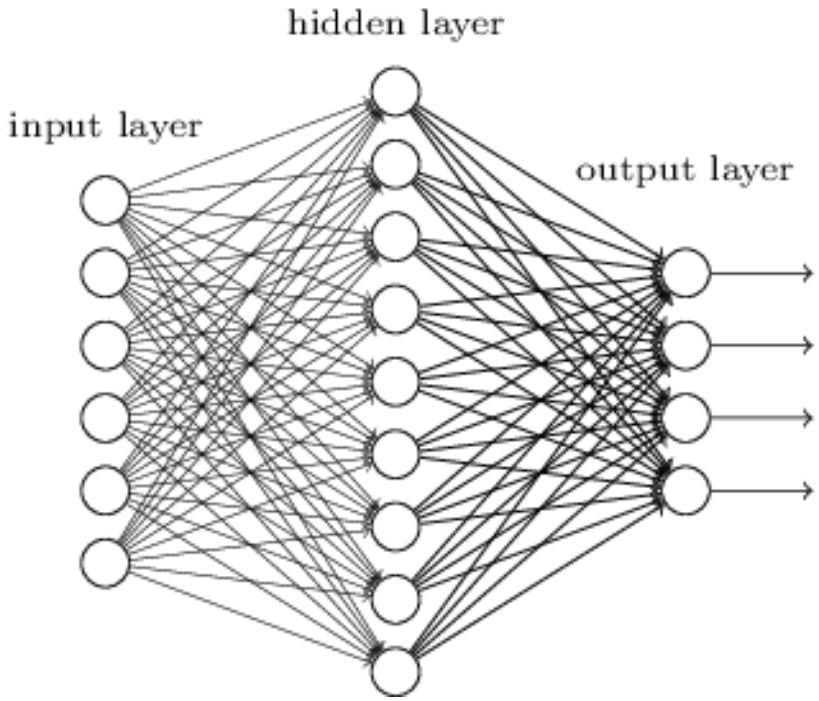


Figura 2.12: Una rete fully-connected [22].

da allenamento e η è il tasso di apprendimento specifico della rete. In tal senso l'algoritmo di backpropagation può essere descritto efficacemente tramite questi passaggi:

1. inizializza un vettore dei pesi W, con pesi random in [-0.05,0.05];
2. calcola $v_{t,i} = \sigma(< w_j^{(t)}, v^{(t-1)} >)$, dove σ è la funzione di attivazione del neurone, w è il suo peso e $v^{(t-1)}$ è l'insieme dei neuroni del layer $t-1$
3. calcola la sensibilità $\delta^{(t)}$ come $\frac{\partial L}{\partial x^{(t)}}$, dove x è il vettore degli input del layer t;
4. aggiorna i pesi per ogni i, j, t utilizzando SGD con la seguente formula:

$$w_{ij}^{(t)} = w_{ij}^{(t)} - \eta v_{t-1,i} \delta_j^{(t)}$$

2.7 Ipotesi di lavoro

Il presente lavoro utilizzerà questi metodi per provare ad analizzare i diversi impatti dei vari aspetti del capitale umano sulla sopravvivenza ed il successo delle startup. In particolare si analizzeranno le seguenti ipotesi:

1. i titoli di studio conseguiti influenzano positivamente il successo della startup;
2. una formazione accademica in un'università prestigiosa ha influenza positiva sul successo della startup;
3. eventuali esperienze lavorative hanno un'influenza positiva sul successo della startup;
4. eventuali precedenti esperienze di fondazione di aziende hanno un'influenza positiva sul successo della startup.

La successiva verifica delle ipotesi qui presenti sarà trattata nella sezione 5.4, in cui verrà discusso anche l'apporto delle singole feature sulla predizione.

Capitolo 3

Dataset

3.1 Estrazione ed elaborazione dei dati

Il dataset a disposizione prende in esame una estrazione dal database di Crunchbase di alcune imprese statunitensi e dei relativi fondatori. L'estrazione, effettuata dal dott. Francesco Ferrati dell'Università di Padova, consta di 10.203 imprese e 18.227 persone. Tuttavia, ai fini di questo lavoro sono state selezionati in maniera randomica 8731 profili di persone e le corrispettive 4753 imprese, questo a causa di limitazioni nel meccanismo che verrà illustrato in seguito.

I dati forniti consistono in due file in formato CSV: *"df_people_organization_status_year"*, illustrato in fig.3.1, e *"df_people_organization"*, illustrato in fig.3.5, contenenti rispettivamente i dati delle imprese e quelli delle persone.

j_uuid	p_uuid	p_firstName	p_lastName	o_uuid	o_name	o_status	o_foundedOnYear
abd64627-02bc-4f3€	41d5f04c-00c7-46e€	Roger	Thomas	0001eae7-077d-4d0	Workspace Property ˆ	operating	2015
84d317ec-ab9f-4332	ce673d04-4320-463	Thomas A.	Rizk	0001eae7-077d-4d0	Workspace Property ˆ	operating	2015
929c9631-622d-1bc	302194b3-87f5-cf4e	Deward	Manzer	000607fc-cea0-535c	BioVigilant Systems	acquired	2005
d30d129d-e14d-e0b	bd63eb82-77ad-052	Changbin	Liu	000cff2d-58df-0a9a-	Termaxia	operating	2015
89face7b-477d-0ccf	3e7615e8-db66-a87	Boon Thau	Loo	000cff2d-58df-0a9a-	Termaxia	operating	2015
1f9d7a57-a6e3-ee3c	31aba383-0d43-bd7	Clint	Rosenblatt	00107fd1-b65b-85cc	CloudAptitude	closed	2012
aa5477af-8718-7d9e	7da5b13e-a536-b92	Michael	Cham	001a1846-f2b2-c3e4	BlenderHouse	operating	2008
60aec39b-a575-7dc	6c2419c4-06a8-0f9c	Jason	Wilburn	001a1846-f2b2-c3e4	BlenderHouse	operating	2008
bf43ce8e-953e-896f	4b615d81-3d94-ac0	Jessica	Hamilton	002194d5-e7cf-cc7E	Smart Lanes	closed	2013
4979114b-b186-d2c	1af30bc7-d7db-52bc	Stephen	Haden	002194d5-e7cf-cc7E	Smart Lanes	closed	2013
802f2751-a592-6b02	592cfc30-97ba-fa5d	Michael	Jansen	00290ee1-0419-336	Satellier	closed	2000
e55d8622-c4ea-565	eadf7345-9417-7c7€	Nick	Carson	002d0ce1-bc3b-36d	PAKIBLE	operating	2014
e1df43fa-9adc-2831	298fd473-51da-aab2	Kimberly	Noonan	00363f9d-0ff0-02ff-e	WindMIL Therapeuti	operating	2015
11858198-d6db-632	35edfce8-ddae-08bb	Ivan	Borrello	00363f9d-0ff0-02ff-e	WindMIL Therapeuti	operating	2015
966a4ddc-645a-aa4	3586b66d-9a27-57b	Raj	Singh	003d551e-f6a6-522C	Comstellar Technolc	closed	2000
6cc7fc13-540f-42cc	881e319d-c00b-460	Tyler	McIntyre	004c2437-53fc-ca03	Novo	operating	2016
70b52a11-efd9-e17c	c0e8400d-18f1-6625	Liz	Keyser	004ea653-3152-15e	MynewMD	operating	2012
027b87ed-d679-66c	54b08f30-d6fe-0825	Chris	Betti	004ea653-3152-15e	MynewMD	operating	2012
e71639a2-d91b-c62	c424dce1-c0f6-71c1	Michael	Kisch	00572f97-a71a-912f	Soundhawk Corporat	closed	2011
604b4a67-efbc-43d8	597be415-0547-1c9	Rodney	Perkins	00572f97-a71a-912f	Soundhawk Corporat	closed	2011
e7389f47-6895-8b6c	f7263bac-648a-df18	Darrin	Murriner	005c35c4-32f5-b0c5	Stockpilz	operating	2014
849f8858-c569-52db	ce885262-5c62-854	David	Barnes	005c35c4-32f5-b0c5	Stockpilz	operating	2014
ba1ebe6d-a5b5-45c	a79e124f-4c18-1e11	Ray	Wang	005d75dc-a3d9-e51	Constellation Resear	operating	2010

Figura 3.1: Una parte del file *"df_people_organization_status_year"*

Nel primo file le imprese sono startup fondate dal 2000 ad oggi e i campi estratti sono:

- **j_uuid**: contentente un codice alfanumerico univoco per ogni ruolo assunto dalla persona in un'impresa;
- **p_uuid**: il codice identificativo della persona;
- **p_firstName**: il nome della persona;
- **p_lastName**: il cognome della persona;
- **o_uuid**: il codice identificativo dell'impresa;
- **o_name**: il nome dell'impresa;

- **o_status**: lo "status" dell'impresa che può essere di quattro tipi: *acquired* se la società è stata acquisita, *operating* se è ancora attiva, *ipo* se ha fatto una *IPO (Initial Public Offer)* e *closed* se ha chiuso;
- **o_foundedOnYear**: l'anno di fondazione dell'impresa.

Da una prima analisi sul dataset fornito delle 10.203 sono risultate:

- 1818 acquired
- 881 closed
- 84 ipo
- 7420 operating

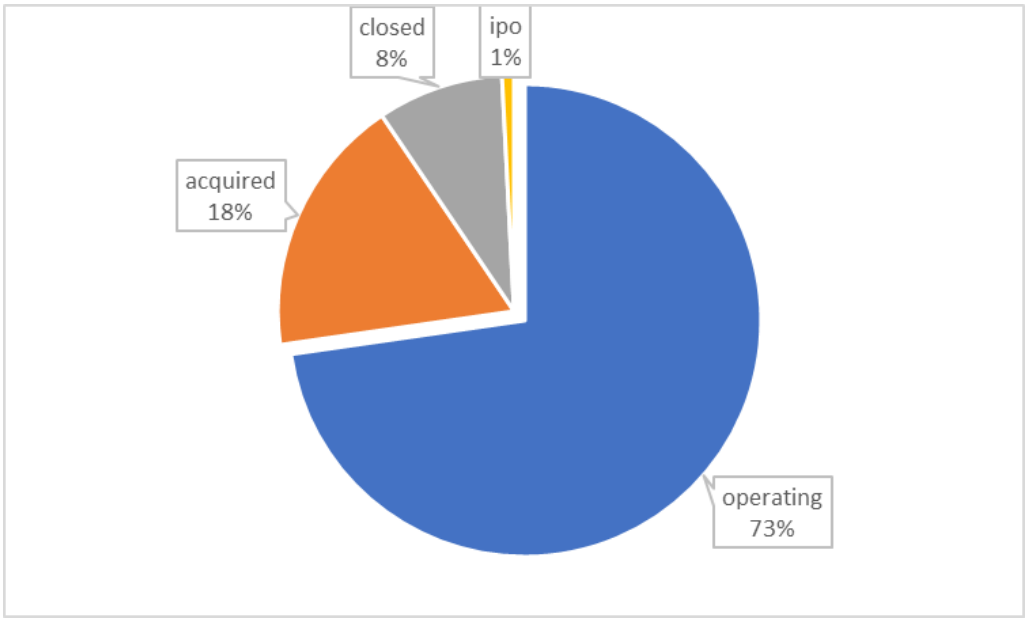


Figura 3.2: La statistica riguardante lo status delle imprese nel file *"df_people_organization_status_year"*

I dati a disposizione, tratti come detto precedentemente da Crunchbase, denotano un grande squilibrio nelle startup prese in esame: solamente l'8% risulta infatti *closed*. Questo fatto dipende principalmente dalla struttura del portale: infatti Crunchbase ottiene i dati dagli stessi founder che iscrivono la loro impresa alla piattaforma per cercare finanziamenti. Pertanto, il database disponibile risulta essere limitato nelle entry relative ad imprese chiuse, infatti spesso gli stessi fondatori chiudono la pagina relativa alla loro startup, rendendo di fatto non più disponibili i dati di queste imprese.

Inoltre, come si può notare dalla tab. 3.1, la maggior parte delle startup in cui le imprese *operating* sono meno del 50% del totale delle fondate in quell'anno sono relativi al periodo 2000-2007. Dunque, questo lasso di tempo ben si presta per le analisi che verranno effettuate successivamente, avendo dei risultati definitivi per la maggior parte delle imprese.

In particolare è possibile effettuare una prima analisi riguardo le medie e le deviazioni standard del numero di imprese fondate per anno e dei vari possibili *status*. I risultati di questa analisi, visibili in fig.3.5, mostrano come i dati siano estremamente volatili, in quanto ad esempio le imprese *operating* siano, in media, il 66% di quelle nate nello specifico anno ma la relativa deviazione standard è del 22%. In definitiva si può notare, già nella tab.3.1, come gli status individuati da Crunchbase varino in maniera importante, soprattutto dopo il 2007, in concomitanza con il notevole incremento di startup registrate al portale.

CLASSES:	TOTAL FOR YEAR	OPERATING	ACQUIRED	CLOSED	IPO
2000	220	38%	45%	12%	5%
2001	129	42%	43%	10%	5%
2002	143	46%	43%	8%	3%
2003	183	44%	45%	8%	2%
2004	208	43%	44%	12%	2%
2005	253	41%	44%	15%	0%
2006	343	48%	39%	13%	1%
2007	354	46%	37%	15%	2%
2008	350	63%	26%	11%	1%
2009	460	62%	29%	9%	1%
2010	686	64%	21%	14%	1%
2011	867	68%	17%	14%	0%
2012	1099	75%	16%	8%	1%
2013	1135	75%	13%	11%	1%
2014	1149	83%	9%	7%	1%
2015	1071	89%	6%	5%	1%
2016	818	95%	3%	2%	0%
2017	569	99%	1%	0%	0%
2018	164	98%	2%	0%	1%
2019	2	100%	0%	0%	0%

Tabella 3.1: La tabella riassuntiva degli status delle imprese suddivise per anno di fondazione

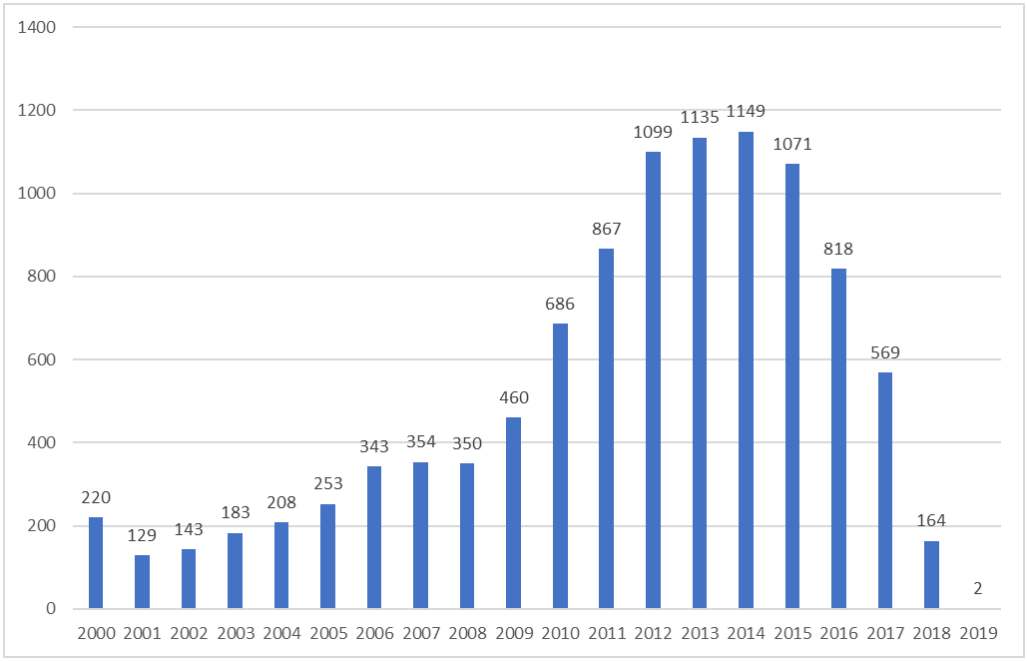


Figura 3.3: Il numero di imprese fondate per anno nel file *"df_people_organization_status_year"*

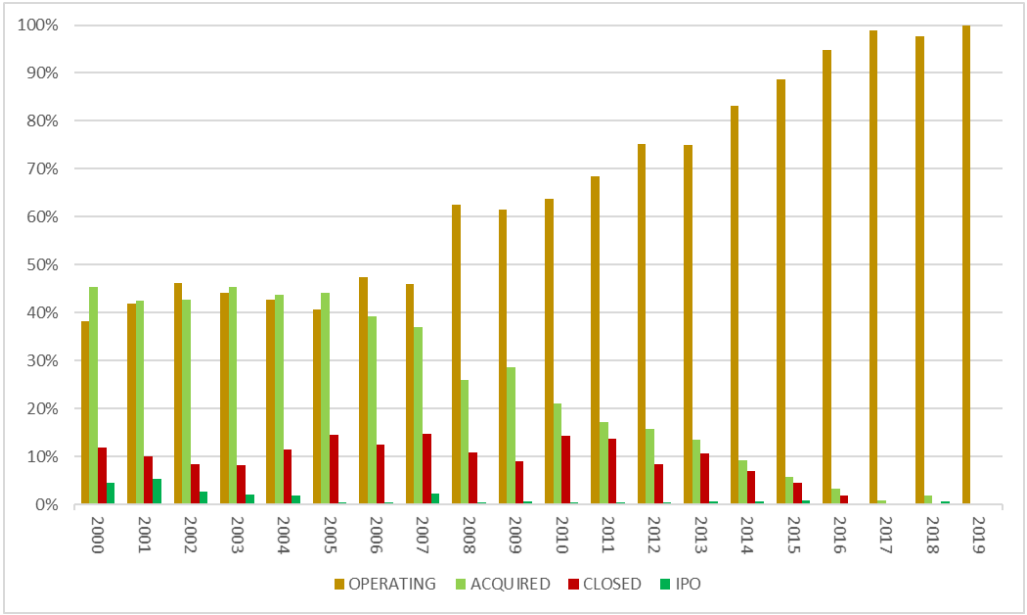


Figura 3.4: Le imprese fondate per anno nel file *"df_people_organization_status_year"*, suddivise per *status*

Nel secondo file, invece, sono contenuti i dati delle persone, in particolare sono stati selezionati questi campi:

- **p_uuid**: il codice identificativo della persona;
- **p_firstName**: il nome della persona;
- **p_lastName**: il cognome della persona;
- **o_name**: il nome dell'impresa;

p_uuid	p_firstName	p_lastName	o_name
ef37d30c-0f20-a314-	Yi	Xiao	Qyer.com
bebbcd40-fef5-6791-	Maya	Azoulay	lool ventures
823eacee-5759-d73:	Wei	Cheng	lool ventures
2dffa4b5-be5f-319b-	Csaba	Kakosy	Talk-A-Bot
b1eccfce-b009-4d3b	Dino	Lauria Ubatuba de F	Bifrost Wealth
e88fa118-7a8f-408a-	Martina	Lofqvist	Bifrost Wealth
d6e01c7e-c6da-4c3:	Zoe	Balek	Bifrost Wealth
a1de7d9b-cba5-454	Viktor	Torstenius	Bifrost Wealth
b24cba3a-8ee4-4d6-	Mike	Polce	M.A. Polce
d56be6ea-98af-4365	Breffney O	Dowling Keane	FruitCubed
d4d14338-6823-40e:	Paul	Monaghan	FOOD-X
ad8c92c4-6c25-4b2-	Isabella	Fantini	FOOD-X
a882cabf-b3fb-4379-	Terry	Romero	FOOD-X

Figura 3.5: Una parte del file *"df_people_organization"*

Durante la prima fase di questo studio, è stata effettuata l'integrazione delle informazioni già presenti con quelle raccolte successivamente: la formazione accademica e le esperienze lavorative.

, per fare ciò si è scelta come fonte il social network professionale LinkedIn. Come già detto in precedenza, l'indagine principale riguarda la formazione accademica e lavorativa, in particolare si è scelto di includere nell'analisi le seguenti voci:

- titolo di studio:
- grado di istruzione, ossia il livello della formazione, suddiviso quindi tra diplomati, laureati e e dottori di ricerca;
- campo di studio, inteso come ambito nel quale la persona ha studiato durante la propria carriera accademica;
- qualità della formazione accademica, intesa come l'esperienza di studi in università prestigiose;
- posizioni lavorative ricoperte in carriera, sia come numero di esperienze che come grado raggiunto;
- durata della carriera lavorativa e delle varie esperienze;
- campo lavorativo, inteso come l'ambito in cui la persona ha lavorato;
- eventuali altre esperienze di fondazione di startup/imprese;

Invece, per quanto riguarda la qualità della formazione accademica si è scelto di utilizzare la classifica ARWU [24], pubblicata annualmente dall'Università di Shanghai, in quanto completa delle migliori 500 atenei mondiali e per l'utilizzo dell'indicatore PCP (*Per Capita Perfomance*).

La classifica è stilata sulla base di cinque indicatori principali:

- **Alumni**, ossia il numero di ex-studenti che hanno vinto premi Nobel e/o medaglie Fields. Ogni assegnazione viene pesata in relazione all'anno di conseguimento, dando un valore maggiore ai premi più recenti;
- **Award**, ovvero il numero del personale accademico che ha vinto uno fra i premi Nobel per la Fisica, Chimica, Medicina ed Economia o la medaglia Fields per la Matematica. L'istituzione alla quale viene assegnato il punteggio è quella di cui faceva parte il vincitore al momento del successo. Allo stesso modo dell'indicatore Alumni il valore è pesato in base al periodo di vincita dei vari premi;
- **HiCi (*Highly Cited Researchers*)**: indice ricavato dalla ricerca di Clarivate Analytics [25] in cui sono selezionati i ricercatori più citati a livello mondiale. Per la classifica ARWU 2019, che è stata utilizzata in questo lavoro, si considera la lista di Clarivate pubblicata nel dicembre del 2018;
- **N&S (*Nature and Science*)** considera, invece, il numero di articoli pubblicati sulle riviste *Nature* e *Science* nei quattro anni precedenti la classifica (per il 2019 si considera il quadriennio 2014-2018). Il punteggio è pesato in base al ruolo avuto nella pubblicazione (autore principale o meno)
- **Pub**, ossia il totale delle pubblicazioni, classificate come "articolo", indicizzate da SCI (Science Citation Index Expand) e SSCI (Social Science Citation Index) nel 2018.

Questi cinque indicatori sono utilizzati nel calcolo del PCP: esso è infatti dato dal rapporto fra la media pesata degli indicatori e il numero di personale accademico dell'università in esame. In questo lavoro verrà utilizzato solamente il PCP in quanto considerato omnicomprensivo e bilanciato rispetto alla grandezza delle singole università (e dei loro fondi).

3.1.1 Raccolta delle informazioni selezionate

A partire dalle informazioni contenute nel file dfpeopleorganization si è integrato, tramite una ricerca automatizzata su Google, l'URI (*Uniform Resource Identifier*) relativo ai profili delle persone presenti nel dataset.

Per la raccolta dei dati necessari si è utilizzata la libreria Linkedin-API [26] in modo da permettere una raccolta massiva delle informazioni richieste. L'output di questa libreria, visualizzabile in fig. 3.6, permette l'estrazione del profilo della persona selezionata in un file JSON. A partire da questi dati si è scelto di selezionare i campi:

- **firstName**
- **lastName**
- **locationName**: località attuale dell'utente;
- **education**, per cui per ogni entry sono stati memorizzati:
  - **schoolName**: nome dell'istituto;
  - **degreeName**: nome del titolo di studio:
  - **fieldsOfStudy**: campo di studio, inserito dall'utente;
  - **timePeriod**, selezionando **endDate**, se presente, e **startDate**
- **experience**, per cui per ogni entry sono stati memorizzati:
  - **companyName**: nome dell'impresa;

  - **title**: posizione ricoperta;
  - **industries**: campo lavorativo;
  - **timePeriod**, selezionando **endDate**, se presente, e **startDate**
  - **locationName**

- **skills**: le competenze inserite dall'utente.


```
{ ...
 'industryName':'Information Technology & Services',
 'lastName':'Brigo',
 'locationName':'Rovigo Area, Italy',
 ...
 'firstName':'Francesco',
'location':{
    'basicLocation':
    'preferredGeoPlace':'urn:li:fs_region:(it,8868)'
 },
 'experience':[
 ],
 'skills':[
    {
      'standardizedSkillUrn':'urn:li:fs_miniSkill:147',
      'name':'Java',
      'standardizedSkill':{
        'entityUrn':'urn:li:fs_miniSkill:147',
        'name':'Java'
      }
    },
    ...
    {
      'name':'SQL'
    }
 ],
 'education':[
    {   ...
       'schoolName':'Università degli Studi di Padova',
       ...
      },
      'timePeriod':{
        'endDate':{
          'year':2019
        },
        'startDate':{
          'year':2017
        }
      },
      'fieldOfStudyUrn':'urn:li:fs_fieldOfStudy:100347',
      'degreeName':'Laurea Magistrale  LM',
      'schoolName':'Università degli Studi di Padova',
      'fieldOfStudy':'Ingegneria informatica',
      ...
    },
    {   ...
       'schoolName':'Università degli Studi di Padova',
       ...
      },
      'grade':'90/110',
      'timePeriod':{
        'endDate':{
          'year':2017
        },
        'startDate':{
          'year':2013
        }
      },
      ...
      'degreeName':'Laurea triennale',
      'schoolName':'Università degli Studi di Padova',
      'fieldOfStudy':'Ingegneria informatica',
    },
 ]
}
```


Figura 3.6: Una parte del file JSON ottenuto come output dalla libreria [26]

Il risultato, visibile in fig.3.7, è stato memorizzato in un file CSV. Una volta ottenute queste informazioni si è passato alla cosiddetta *feature engineering*, argomento che verrà affrontato nella prossima sezione.

Katerina	Stroponiati	San Francisco, California	|UC Berkeley College of Environmental Design||Technical University of Crete|	|None||MEng|	|Architecture, Landscape Architecture, and Sustainable City Planning.||Management and	||2007-2002
Bruce	Pla	San Francisco, California	|Y Combinator||Academy of Art University|	|None||Bachelor's degree|	|Entrepreneurship/Entrepreneurial Studies||Industrial and Product Design|	|2016-2015|2012-2009
Nelson	Vazquez	San Francisco, California	|University of Puerto Rico-Mayaguez|	|Bachelor's Degree|	|Electrical and Electronics Engineering|	|2010-2003
Eric	Sanchez	San Francisco, California	|YCombinator||Academy of Art University||University of Puerto Rico-Mayaguez|	|Master's Degree||Master of Fine Arts (MFA)||Bachelor of Science (BS)|	|None||Industrial and Product Design||Electrical Engineering - Control Systems|	|2016-2016|2016-2013|2010-2004
Brian	Bennett	New York, New York	|Y Combinator||University of Rochester|	|None||B.S.|	|None||Optical Engineering. Minor, Mathematics|	|2015-2015|2008-2004
Paul	Fulton	San Francisco Bay Area	|Michigan Technological University|	|BSEE|	|Computer Engineering|	|NULL
Nicholas R.	Karp	La Jolla, California	|Princeton University||Rutgers, The State University of New Jersey-Newark||Tools & Technologies|	|AB||MBA||None|	|Classics||Professional Accounting||None|	|1984-1980|2010-2009|NULL
Eric Lee	Smith	Albuquerque, New Mexico	|Pratt Institute|	|BFA|	|Fine Art & Photography|	|1982-1978

Figura 3.7: Una parte del file contenente tutte le informazioni riguardanti i founder

3.2 Feature Engineering

Una volta in possesso delle informazioni sulle persone contenute nel dataset, si è deciso di analizzare e convertire ogni feature in valori numerici. In particolare, si è scelto di mantenere le informazioni standardizzate dalla piattaforma , scartando, di conseguenza, le informazioni relative alle *skill* ed al campo di studi. Il dataset è stato quindi modificato come riassunto in tab.3.2.

Una spiegazione integrativa è necessaria per i campi *ed_ tech*, *ed_ business*,*ed_ other*, *j_ serialEnt* e *j_ moveToUSA*.

Infatti, nei primi 3 casi è stato effettuato un controllo automatico dei vari titoli di studio presente nel curriculum in base all'indice disponibile in [27] cercando nell'elenco dei titoli accademici conseguiti sia gli acronimi a disposizione sia l'intero diploma. In tal modo si è riusciti a risalire ad una prima caratterizzazione del campo di studio, come voluto inizialmente.

Per quanto riguarda il campo *ed_ ranking* si è proceduto, sulla base di quanto detto precedentemente riguardo il ranking ARWU, ad effettuare una distinzione sulla base del grado di istruzione ottenuto dalla specifica università: infatti si è attribuito un peso pari a 3 per il titolo di dottorato, 2 per il master e 1 per il bachelor.

$$uRank = 3 \cdot PHDuRank + 2 \cdot MasteruRank + BacheloruRank \tag{3.2.1}$$

Inoltre, sono state utilizzate le informazioni contenute nel dataset originario per trovare le persone che sono considerate *serial enterpreneur*, ossia se hanno fondato più di un'impresa.

Infine, è stato inserito il campo *j_ moveToUSA* che rappresenta lo spostamento della singola persona negli USA per fondare la startup. Per ottenere l'evidenza di questo spostamento sono state analizzate tutte le posizioni lavorative e di formazione accademica precedente per verificare l'effettivo trasferimento.

Come detto in precedenza, l'effettivo dataset utilizzato in questo lavoro considera una selezione randomica del dataset originario, a causa di limitazioni delle piattaforma da cui vengono presi i dati. Dalle elaborazioni ottenute sono quindi risultati 8731 profili di persone e 4753 imprese.

Una prima analisi effettuata è quella chiamata *statistica descrittiva*, da cui risulta che il dataset è così composto:

Nome del campo	Valori	Spiegazione del campo	Metodologia di estrazione
ed_phd	{0,1}	Presenza o meno del diploma di dottorato	Dall'output della libreria, campo education
ed_master	{0,1}	Presenza o meno della laurea magistrale	
ed_bachelor	{0,1}	Presenza o meno della laurea	
ed_ranking	[0,1]	Ranking delle università frequentate (se nella top 500 di ARWU)	Dal ranking ARWU, pesato per grado del titolo: ed_ranking=3*rankingPHD+ 2*rankingMaster+ 1*rankingBachelor
ed_tech	{0,1}	Campo di studi tecnico/tecnologico	Dall'elenco disponibile qui sono stati verificati titolo di studio
ed_business	{0,1}	Campo di studi economico/business/finanziario	
ed_other	{0,1}	Tutti i campi di studi non separatti precedentemente	
j_nYears	Integer	Numero di anni di carriera lavorativa	Dall'output della libreria, campo experience
j_nPrevJobs	Integer	Numero di posizioni lavorative ricoperte, escludendo quella della startup in esame	
j_serialEnt	{0,1}	Il campo è pari ad 1 se la persona ha fondato altre startup	Ricerca per p_uuid nel file originale df_people_organization.csv
j_moveToUSA	{0,1}	Il campo è pari ad 1 se la persona si è spostata negli USA per fondare la startup	Ogni startup nel file originale è fondata negli USA, il controllo è stato fatto sulle esperienze lavorative precedenti

Tabella 3.2: Tabella riassuntiva delle elaborazioni fatte a partire dai curriculum

- come visibile in fig.3.8,i gradi di istruzione più rappresentati sono il *bachelor* (33%) e il *master*(27%), tuttavia un terzo delle persone presenti non ha una formazione universitaria (29%). Solamente l'11% del campione possiede un dottorato;
- sebbene il gruppo più numeroso di lauree sia quello di stampo tecnico/tecnologico, il numero di laureati in discipline non tecnico/economiche è preponderante. Come ben visibile nella fig.3.9, il 74% del campione analizzato ha un titolo di studio non relativo ad ambiti economici, rappresentati dal 4%, nè tecnici, il rimanente 22%;
- il 19% delle persone contenute nel dataset non sono americane;
- solamente il 10% del campione ha studiato in una delle università top-500 secondo ARWU;
- il numero medio di posizioni lavorative ricoperte prima di fondare la startup è superiore a 3 (precisamente 3,49), mentre la durata media della carriera è di 11 anni circa.
- la variabilità di questi valori, calcolata mediante la deviazione standard, è alta per la durata media della carriera ($\sigma = 24,61$), mentre per il numero di posizione ricoperte è più bassa ($\sigma = 1,08$).

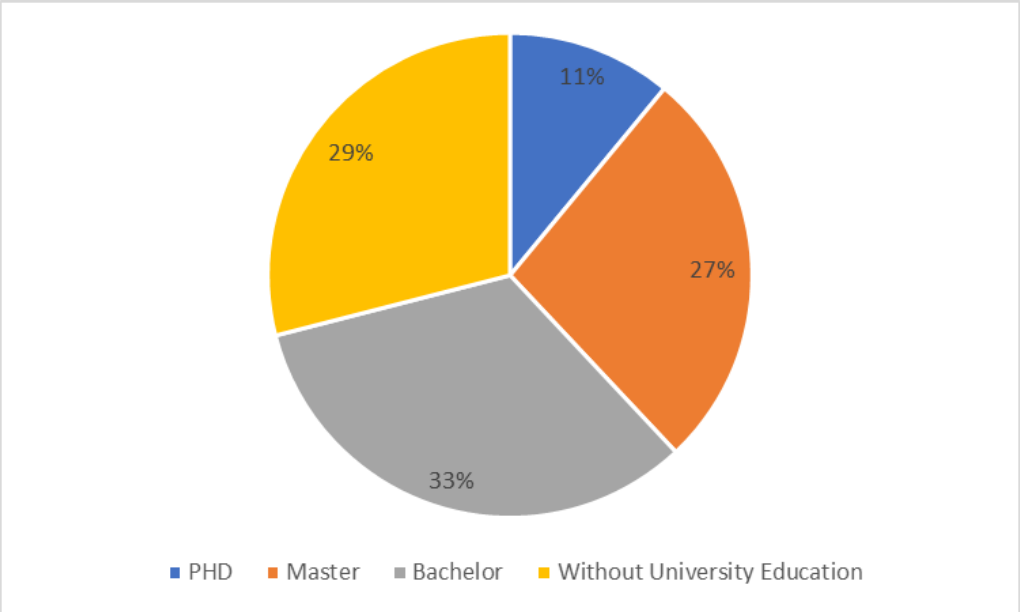


Figura 3.8: I gradi di istruzione per i profili selezionati

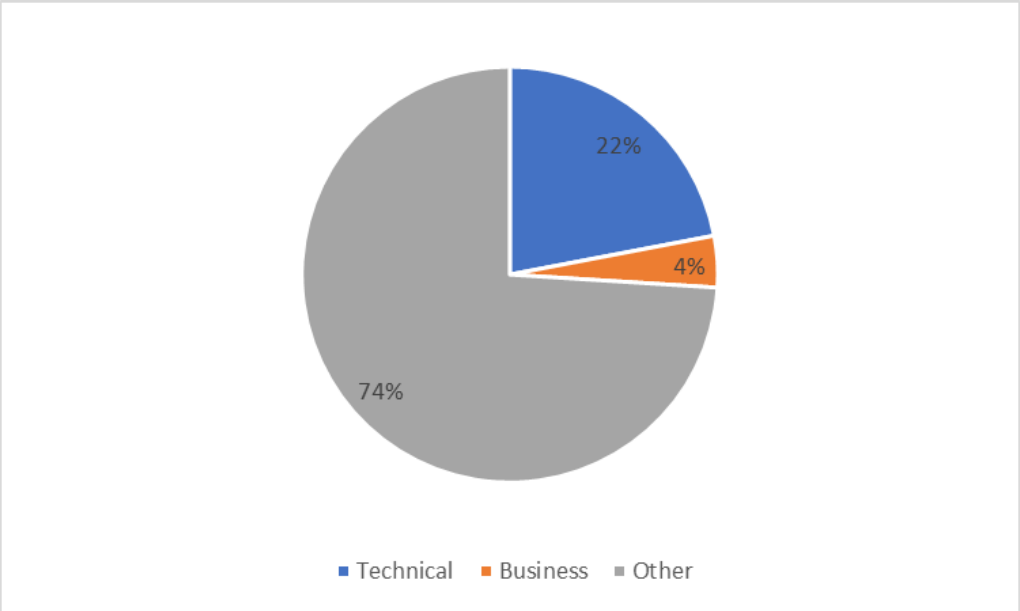


Figura 3.9: I campi nei quali i founders hanno studiato

3.3 Pulitura del dataset

La fase successiva prevede la pulitura dei dati ottenuti e la loro trasformazione nei team che verranno poi analizzati complessivamente. Per delineare un profilo del team si è utilizzato il seguente approccio, suddiviso in step successivi:

1. sono state selezionate le imprese con status definiti (closed, acquired o ipo) e le imprese operating fondate prima del 2007;
2. si è deciso a che classe assegnare le varie imprese:
$$status = \begin{cases} 0, & \text{se lo status è } closed \\ 1, & \text{altrimenti} \end{cases}$$
3. sono stati creati i profili delle startup sommando le singole caratteristiche pesando il loro apporto in base al numero di membri del team dei founder.

3.3.1 Pesatura

Secondo alcune ricerche nel campo delle *startup*, fra cui [29], il numero di fondatori impatta sull'effettiva riuscita. Da un'analisi effettuata dal blog specializzato TechCrunch [28] la numerosità del team di startup che vengono considerate di successo si aggira attorno ai 2 membri (precisamente 1,85). Infatti, dall'analisi del database messo a disposizione da Crunchbase Kamps nota come, fra le startup con un *exit* (*ipo/acquired*) la media del numero dei fondatori sia 1,72, ciò significa che più della metà delle startup *ipo/acquired* hanno un solo fondatore. Al fine di valutare più in profondità l'ipotesi della non-necessità di un cofounder per avere successo, vengono analizzate le startup di successo mediante due criteri:

1. ottenimento di più di 10 milioni di dollari di finanziamento;
2. uscita (*ipo/acquired*)

In entrambi i casi il gruppo più numeroso (fra il 46% del caso 1. e il 52% del caso 2.) è quello di imprese fondate da una sola persona. Tuttavia, la statistica più interessante

riguarda i team da tre o più persone, infatti solo una percentuale fra il 22% e 17% riesce ad avere successo. Questi risultati potrebbero dipendere, come detto anche in [29], da un ruolo diverso nelle dinamiche interne al team. È bene sottolineare però che investitori quali Paul Graham di Y Combinator e Dave McClure di 500 Startups riconoscono l'importanza di avere almeno un co-founder come base per la buona riuscita della startup fondata.

Per riflettere questi lavori e queste autorevoli opinioni in cui viene sottolineata l'importanza del team è stata utilizzata una funzione di pesatura ad-hoc. Quest'ultima è stata sviluppata per considerare sia le dinamiche di gruppo all'interno del team, penalizzando quindi i team più numerosi, sia l'influenza positiva del team rispetto al singolo, penalizzando quindi i team di founder singoli rispetto a quelli con almeno un co-founder.

Considerando la legge dei grandi numeri possiamo pensare che la distribuzione della probabilità legata alla numerosità del team della startup tenda ad essere approssimabile ad una distribuzione normale standard. Di conseguenza, nel dataset utilizzato in questo lavoro, sono stati calcolati il numero medio di fondatori per le startup, pari a 1,93, e la deviazione standard, pari invece a 1,36.

Dunque, la pesatura è stata effettuata utilizzando la formula seguente, adattando media μ e deviazione standard σ ai dati a disposizione ($\mu = 1,93$, $\sigma = 1,36$):

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2} \tag{3.3.1}$$

Numero di founder	**Peso della singola feature**
1	0.2321827405367910000
2	0.2929516080983820000
3	0.2152570068266950000
4	0.0921115658729466000
5	0.0229544288364181000
6	0.0033313040591472200
7	0.0002815512854753630
8	0.0000138578613718818
9	0.0000003972193239845
10	0.0000000066307091873
11	0.0000000000644592208
12	0.0000000000003649266

Tabella 3.3: I valori della distribuzione gaussiana calcolati per il numero di founder

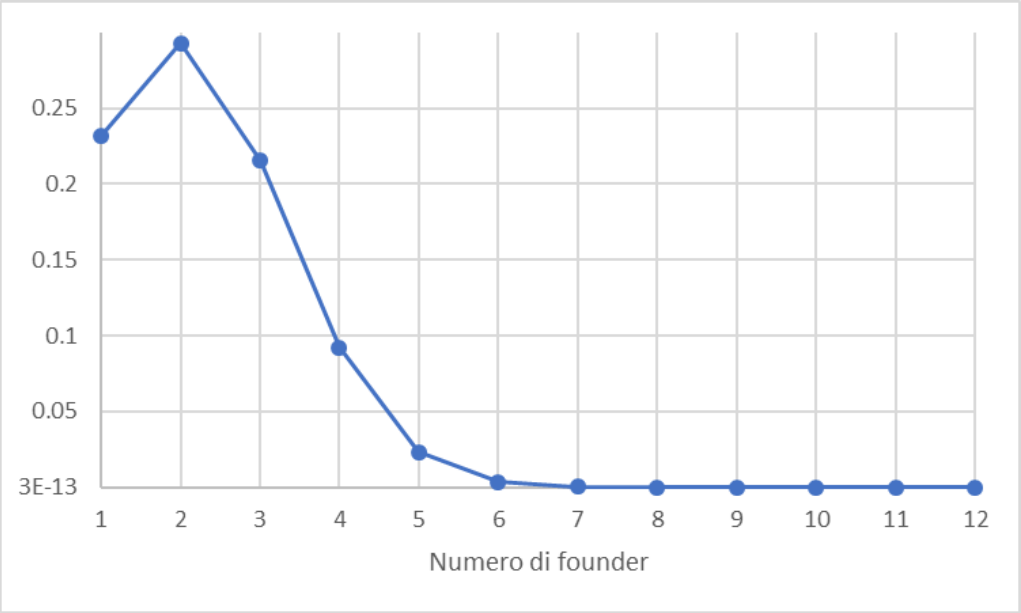


Figura 3.10: Il grafico della gaussiana costruita per la pesatura delle feature

Una volta creati i profili dei team dei founder si è poi proceduto con l'implementazione dei cinque modelli presentati precedentemente, come verrà spiegato nel cap.4.

Capitolo 4

Implementazione dei modelli

In questo capitolo verrà presentata la creazione e l'implementazione dei modelli. In una prima fase verranno spiegate alcune elaborazioni sul dataset al fine di bilanciarlo, mentre successivamente si spiegheranno i modelli, la cui spiegazione teorica è contenuta nel capitolo 2, e le scelte implementative fatte.

4.1 Bilanciamento del dataset

Una volta ottenuto il dataset, ulteriormente ridotto a 1732 aziende, consta di:

- 1339 aziende classificate come "1";
- 393 aziende classificate come "0".

Per ottenere un dataset bilanciato si è quindi fatto ricorso ad una delle tecniche di *oversampling*, ossia di creazione di esempi sintentici sulla base dei record reali contenuti nel dataset.

In questo lavoro si è scelto di utilizzare la tecnica SMOTE *(Synthetic Minority Oversampling Technique)* [30]. Questo metodo permette di creare dei record sintetici a partire dai dati della classe di minoranza. In particolare, per ogni record x_i appartenente alla classe di minoranza vengono presi in considerazione i k-nearest neighbors, ossia i k record della classe di minoranza la cui distanza nei confronti di x_i presente il valore minimo lungo le dimensioni dello spazio vettoriale. Il meccanismo di creazione prevede quindi, il calcolo del nuovo record come segue:

$$x_{new} = x_i + (\hat{x}_i - x_i) \cdot \delta_i \tag{4.1.1}$$

dove $\hat{x}_i$ è uno dei k vicini di x_i e $\delta_i \in [0, 1]$ rappresenta un numero random. In tal modo il record sintetico si troverà sulla retta che collega x_i e $\hat{x}_i$. L'algoritmo, illustrato in fig.4.1, ha come maggior limite quello di generalizzare senza avere una conoscenza dell'intero dataset ma considerando solamente la classe di minoranza.

Algoritmo 2.3 Creazione di un esempio sintetico con SMOTE

1) Trova i k vicini per ciascuna istanza appartenente alla classe di minoranza
2) Seleziona in maniera random il vicino x_j di x_i $(1 \leq j \leq k)$
3) Calcola la differenza tra i valori degli attributi dell'istanza x_i e il vicino x_j: $diff = x_j - x_i$
4) Genera un numero random δ (compreso tra 0 e 1)
5) Crea l'esempio sintetico: $x_{new} = x_i + diff \cdot \delta$

Figura 4.1: L'algoritmo SMOTE

In particolare, il rischio maggiore dato dall'applicazione dell'algoritmo SMOTE è quello della cosiddetta *overgeneralization*, ossia una riduzione delle regioni di competenza delle varie classi, rendendo di fatto difficile una classificazione accurata. Per implementare ed effettuare l'oversampling è stata utilizzata la libreria imblearn [32], di cui viene usata la classe SMOTE.

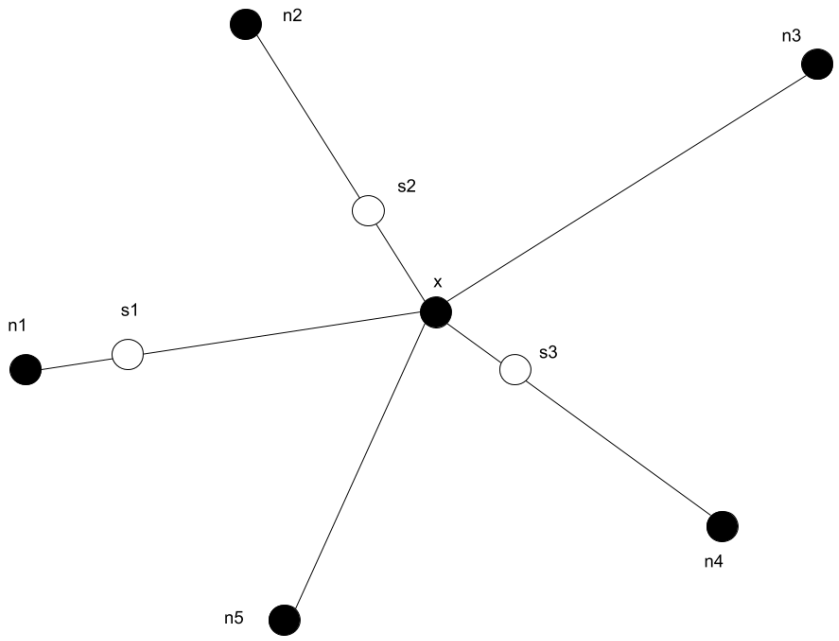


Figura 4.2: Un esempio di applicazione dell'algoritmo SMOTE, in questo caso s_1, s_2, s_3 sono i record sintetici creati

4.2 Parametri dei modelli

Durante la fase di implementazione dei modelli si è utilizzata la libreria *sklearn* [31] ed in particolare le classi:

- LogisticRegression;
- DecisionTreeClassifier;
- KNeighborsClassifier;
- RandomForestClassifier;
- MLPClassifier;

Per ognuno dei metodi scelti si è scelto di utilizzare il metodo *GridSeachCV* in modo da calcolare, iterativamente, il modello migliore a partire da un set possibile di parametri. Nei prossimi paragrafi verranno spiegati più in dettaglio i modelli e le loro caratteristiche mentre la valutazione dei risultati ottenuti sarà oggetto del cap.5. Ognuno dei modelli che verranno presentati è stato valutato sulle metriche AUC, Accuracy, F1, Precision e Recall, mentre il *refitting* è stato effettuato sulla Recall (per il dettaglio sulle metriche utilizzate si rimanda al cap.5, sezione 5.1). Quest'ultima impostazione, nonostante porti generalmente ad avere modelli con una *precision* più bassa, permette di ottenere i modelli con la miglior precisione sui falsi negativi.

4.2.1 Regressione Logistica

Il modello di regressione logistica che verrà utilizzato è così impostato:

- C=1.0: il parametro C indica l'inverso della *regularization strength*. Un valore basso indica una regolarizzazione maggiore, infatti indica un grado di penalizzazione sui singoli valori (più è basso meno sono penalizzati le feature con valori piccoli);

- class_weight=None: il parametro indica eventuali specifici pesi per le diverse classi, in questo caso tutte le classi hanno peso pari a 1;
- dual=False: questo parametro indica la formulazione primale o duale del problema, consigliato a *False* quando, come in questo caso, il numero di osservazioni supera il numero di feature;
- fit_intercept=True: indica l'eventuale uso di una costante, specificata dal parametro intercept, come bias;
- intercept_scaling=1: il parametro indica l'aggiunta di una feature sintetica con valore pari ad 1;
- l1_ratio=None: indica che nessuna penalità di tipo l1 è stata utilizzata;
- max_iter=100: il numero massimo di iterazioni dell'algoritmo;
- multi_class='warn'
- n_jobs=None: numero di CPU utilizzate se il multi_class risulta essere abilitato;
- penalty='l2': la penalizzazione selezionata. In questo caso l2 significa che viene utilizzata una regressione Ridge;
- solver='warn': il tipo di solver selezionato
- tol=0.0001: la tolleranza impostata per i criteri di arresto dell'algoritmo;

4.2.2 Decision Tree

Il modello di albero di decisione è così definito:

- class_weight=None: il parametro indica eventuali specifici per le diverse classi, in questo caso tutte le classi hanno peso pari a 1;
- criterion='gini': indica la funzione per misurare la qualità dei vari split presenti nell'albero. In questo caso è utilizzata la funzione di entropia di Gini [33], rappresentata dalla formula:

$$I_G(p) = 1 - \sum_{i=1}^{J} p_i^2 \tag{4.2.1}$$

  dove J è il numero di classi e p_i la parte di elementi classificati come i.
- max_depth=None: indica la massima profondità dell'albero. In questo caso non risulta impostata, dunque l'albero verrà espanso finché tutte le foglie non saranno pure, ossia con tutti gli elementi appartenenti ad una sola classe, oppure fino a che non contengano meno di min_samples_split esempi;
- max_features=None: indica il massimo numero di feature da considerare per ciascuno split;
- max_leaf_nodes=None: indica il massimo numero di foglie che l'albero può avere, se, come in questo caso, è *None* allora le foglie sono illimitate;
- min_impurity_decrease=0.0: la soglia minima di decrescita dell'impurità per ogni split;
- min_samples_leaf=1: il numero minimo di osservazioni necessarie per creare un nodo foglia;

- min_samples_split=2: il numero minimo di osservazioni utili a rendere necessario uno split del nodo interno;
- presort=False: indica se è stato effettuato un sorting prima di eseguire l'algoritmo per velocizzare la ricerca degli split migliori;
- splitter='best': indica la strategia utilizzata per scegliere lo splitting ad ogni nodo. In questo caso è scelta sempre quella migliore (*approccio greedy*).

4.2.3 K-Neighbors Classifier

Il classificatore basato sull'algoritmo KNN è definito come segue:

- algorithm='auto': la scelta dell'algoritmo è dipendente dai dati, fra le possibilità ci sono BallTree, KDTree [34] e l'approccio brute force;
- leaf_size=30, ossia la grandezza delle foglie passate agli algoritmi BallTree e KDTree;
- metric='minkowski': la metrica di distanza utilizzata, in questo caso è la distanza di Minkowski, il parametro p è dipendente dalla distanza utilizzata;
- metric_params=None, eventuali altre caratteristiche della funzione distanza;
- n_jobs=None: il numero di job paralleli utilizzati per cercare le osservazioni vicine;
- n_neighbors=5: il numero di osservazioni "vicine" da considerare;
- p=2: indica che stiamo utilizzando la distanza euclidea;
- weights='uniform': i pesi utilizzati in fase di predizione. In questo caso sono uniformi per tutte le osservazioni.

4.2.4 Random Forest

Il modello Random Forest è invece definito così:

- bootstrap=True, indica se il *bootstrapping* è utilizzato per costruire la foresta;
- class_weight=None: nessun peso specifico per le classi rappresentate viene impostato;
- criterion='gini': indica la funzione per misurare la qualità dei vari split presenti nell'albero.
- max_depth=None: indica la massima profondità di ogni albero della foresta
- max_features='auto': indica il massimo numero di feature da considerare per ciascuno split
- max_leaf_nodes=None: indica il massimo numero di foglie che ogni l'albero può avere;
- min_impurity_decrease=0.0, come in 4.2.2, indica la soglia minima di decrescita dell'imputrità per effettuare uno split;
- min_impurity_split=None, come in 4.2.2
- min_samples_leaf=1, come in 4.2.2

- min_samples_split=2, come in 4.2.2
- n_estimators='warn', rappresenta il numero di alberi nella foresta
- n_jobs=None, come in 4.2.2
- oob_score=False, indica l'utilizzo o meno di osservazioni *out-of-bag* per stimare l'accuratezza della generalizzazione.

4.2.5 Rete neurale

Per creare la rete neurale si è scelta la classe MLPClassifier che mette a disposizione un modello basato sulla struttura a percettrone. La rete che verrà utilizzata è composta come segue:

- activation='relu', indica la funzione di attivazione. In questo caso è utilizzata la funzione $f(x) = max(0, x)$
- alpha=0.0001: la penalità L2
- batch_size='auto': la grandezza dei minibatch per gli ottimizzatori stocastici
- beta_1=0.9
- beta_2=0.999
- early_stopping=True, indica la possibilità di terminare precocemente l'algoritmo;
- epsilon=1e-08
- hidden_layer_sizes=(20,), indica il numero di neuroni nei vari layer nascosti;
- learning_rate='constant': il learning rate per l'aggiornamento dei pesi;
- learning_rate_init=0.001: il learning rate iniziale;
- max_iter=200: il numero massimo di iterazioni;
- momentum=0.9: il momentum dell'aggiornamento per la discesa del gradiente;
- n_iter_no_change=10, indica il numero massimo di *epoch* per cui non si ha un miglioramento
- nesterovs_momentum=True: l'utilizzo o meno del momentum di Nesterov;
- power_t=0.5, l'esponente per l'inverso del learning rate.

Capitolo 5

Risultati sperimentali

In questo capitolo verranno presentati i risultati della classificazione e verrà identificato il modello più performante per la predizione del successo delle startup. Infine verranno presentati e discussi, in relazione alle ipotesi di lavoro formulate nel paragrafo 2.7, i risultati legati alla feature importance per il modello migliore.

Nelle prossime sezioni verranno approfondite le metriche scelte per la valutazione e la validazione dei modelli.

5.1 Metriche di valutazione

Come già menzionato nel capitolo precedente sono state utilizzate le seguenti metriche per valutare i modelli di Machine Learning proposti in questo lavoro. Nelle prossime sezioni verranno introdotte le metriche e la loro importanza nella valutazione degli algoritmi di Machine Learning.

5.1.1 Accuracy

L'accuratezza è una delle possibili metriche per valutare la bontà di un modello di Machine Learning. Come evidenziato dall'equazione 5.1.1, l'accuratezza rappresenta la percentuale di classificazioni correttamente etichettate sul numero di predizione fatte. Per la classificazione binaria, come in questo caso, si calcola come:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1.1}$$

dove TP sono i True Positive, TN i True Negative, FP i False Positive e FN i False Negative.

5.1.2 Precision

La precisione indica, come mostrato nell'equazione 5.1.2, la proporzione di classificazione positive che sono effettivamente corrette.

$$Precision = \frac{TP}{TP + FP} \tag{5.1.2}$$

5.1.3 Recall

La *recall* (richiamo) indica, invece, la percentuale di osservazioni etichettate positivamente che sono state effettivamente identificate

$$Precision = \frac{TP}{TP + FN} \tag{5.1.3}$$

5.1.4 F1

La metrica F1 combina gli effetti della Precision e della Recall in un unico valore, essa infatti è calcolata come segue:

$$Precision = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{5.1.4}$$

Questa metrica rappresenta una media pesata della precisione e del richiamo, fondendo le due metriche e rappresentando in maniera completa l'accuratezza del modello.

5.1.5 Receiver operating characteristic e Area Under the Curve

La curva ROC (*Receiver operating characteristic*) rappresenta le performance complessive della classificazione utilizzando come parametri TPR (*True Positive Rate* o sensitività) e FPR (*False Positive Rate*), calcolati come segue:

$$TPR = \frac{TP}{TP + FN} \tag{5.1.5}$$

$$FPR = 1 - \text{specificity} = \frac{FP}{FP + TN} \tag{5.1.6}$$

L'utilizzo della ROC, e della misura collegata AUC (*Area Under the Curve*), permette di valutare quanto il modello classifichi correttamente le varie osservazioni. In particolare un buon modello avrà una AUC vicina ad 1 e una ROC che segue l'asse y e il limite superiore, come visibile in fig.5.1. La metrica AUC calcolata come l'area sottesa dalla curva ROC permette di valutare la qualità del modello nel distinguere fra le due classi: in particolare una AUC=0 significa che il modello classifica le osservazioni esattamente l'opposto di quello che dovrebbe.

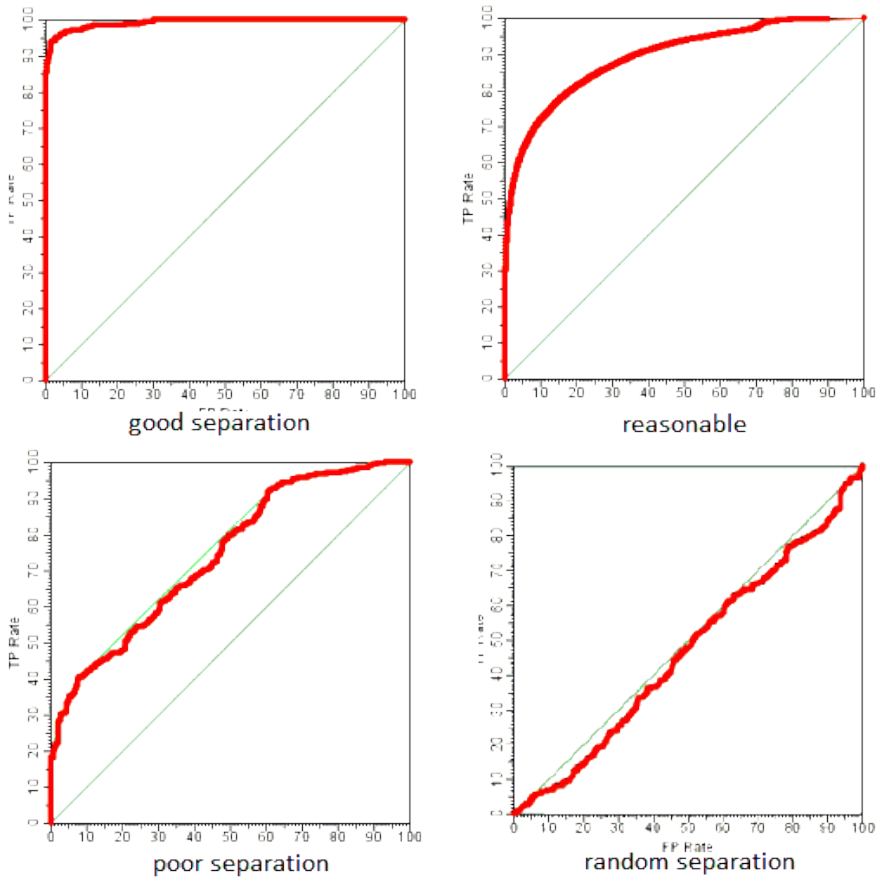


Figura 5.1: Quattro esempi di curva ROC con un'interpretazione, da [35]

5.2 Cross Validation

Per effettuare una efficace validazione si è scelto di utilizzare i metodi classici della cross validation, in particolare sono state vagliate la *K-fold Cross Validation*, la *Stratified Cross Validation* e la validazione *Leave One Out*. Tuttavia, a causa della limitata grandezza del dataset e, di conseguenza, delle poche osservazioni a disposizione si è scelto di utilizzare la stratified cross-validation perché include nel validation set un numero uguale di osservazioni delle due classi. Quindi, evitando di far fallire il task per la presenza nell'insieme di validazione di una sola classe.

La *Stratified K-Fold Cross Validation* è una tecnica di cross-validation per cui i dati vengono riorganizzati in maniera che ogni parte sia una buona rappresentazione dell'intero dataset. In questo modo si è certi che ogni sottoinsieme sia rappresentato dallo stesso numero di osservazioni per ciascuna classe. Un esempio di stratified cross validation può essere visto in fig.5.2.

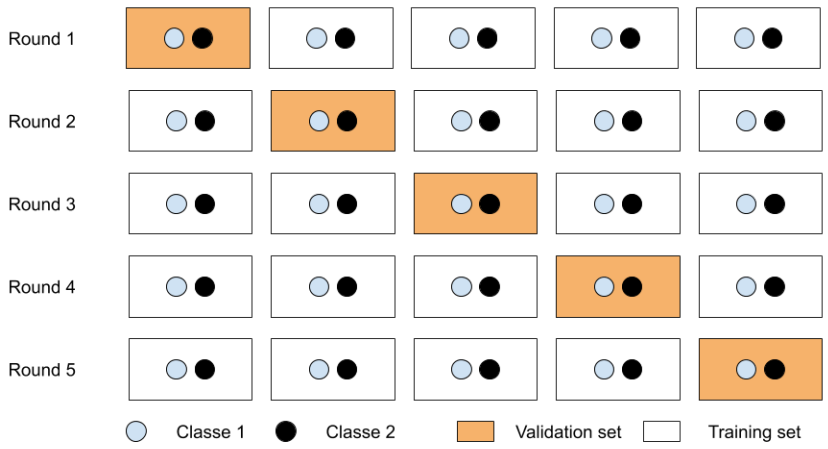


Figura 5.2: Un esempio di stratified k-fold cross validation per una classificazione binaria

5.3 Risultati dei modelli utilizzati

In questa sezione verranno presentati i risultati conseguenti dai modelli illustrati nel capitolo 4 sulla base del dataset creato in questo lavoro. Inizialmente verranno presentati i risultati conseguiti da ogni modello e verranno discussi i punti di forza e le debolezze di ogni modello presentato. Infine, verrà selezionato il modello migliore per il task in esame.

In Tabella 5.1 vengono riportati tutti i valori raggiunti dalle varie metriche per i modelli analizzati. Come è ben visibile della tabella e dalle figure successive il modello che performa meglio sui dati proposti è quello delle Random Forest, ottenendo circa l'80% su tutte le metriche. In particolare, si può evidenziare come la AUC sia pari all'88%, ciò significa che il modello distingue correttamente fra le startup con status 0 e quelle con status 1. Inoltre, come ben visibile anche nella fig.5.3, anche i valori della precisione e della recall ottenuti sono buoni considerando l'obiettivo di questo lavoro. I risultati ottenuti dal modello Random Forest sono confermati anche dal modello Decision Tree, di cui le foreste sono un insieme. Infatti, vi è un incremento del 5,5% per quanto riguarda la precisione, del 16,8% per la recall, del 10,9% per la F1 e del 14,8% per la AUC. Invece, per quanto riguarda i modelli di regressione logistica e di rete neurale (MLPClassifier), i risultati ottenuti sono più bassi degli altri, con un'accuratezza fra il 50 e il 63% e una recall inferiore al 65%. Questi risultati dipendono, nel caso della regressione logistica, dalla correlazione fra le feature selezionate (si pensi ad esempio ai titoli di studio), mentre, nel caso della rete neurale, dipendono, in parte, dal ristretto numero di osservazioni a disposizione nel training set. Dunque, i risultati ottenuti portano a scartare questi modelli per l'analisi delle ipotesi fatte al capitolo 2. Infine, il modello basato sull'algoritmo K-Nearest Neighbors, pur ottenendo dei risultati comparabili con l'albero di decisione, non risulta il migliore del lotto e viene dunque scartato a favore del modello Random Forest.

Modello	**Accuracy**	**Precision**	**Recall**	**F1**	**AUC**
Logistic Regression	60,83%	62,78%	52,87%	57,16%	64,95%
Decision Tree	73,49%	78,92%	68,93%	72,76%	76,65%
Random Forest	79,84%	83,23%	80,50%	80,68%	88,00%
MLP Classifier	58,81%	59,49%	55,11%	56,96%	60,02%
K-Nearest Neighbors	74,24%	78,90%	69,45%	73,30%	75,74%

Tabella 5.1: I risultati dei modelli sulle varie metriche

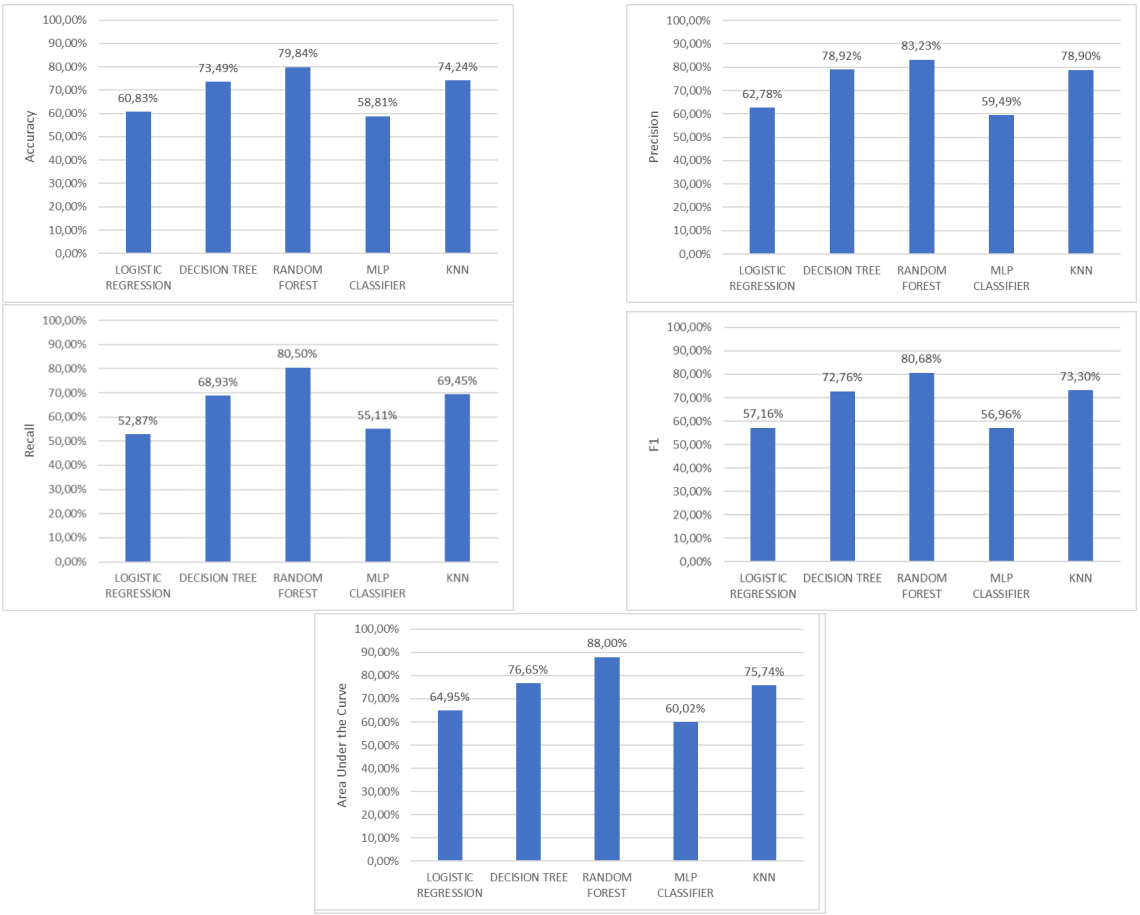


Figura 5.3: I risultati presenti in tab.5.1 per le cinque metriche analizzate

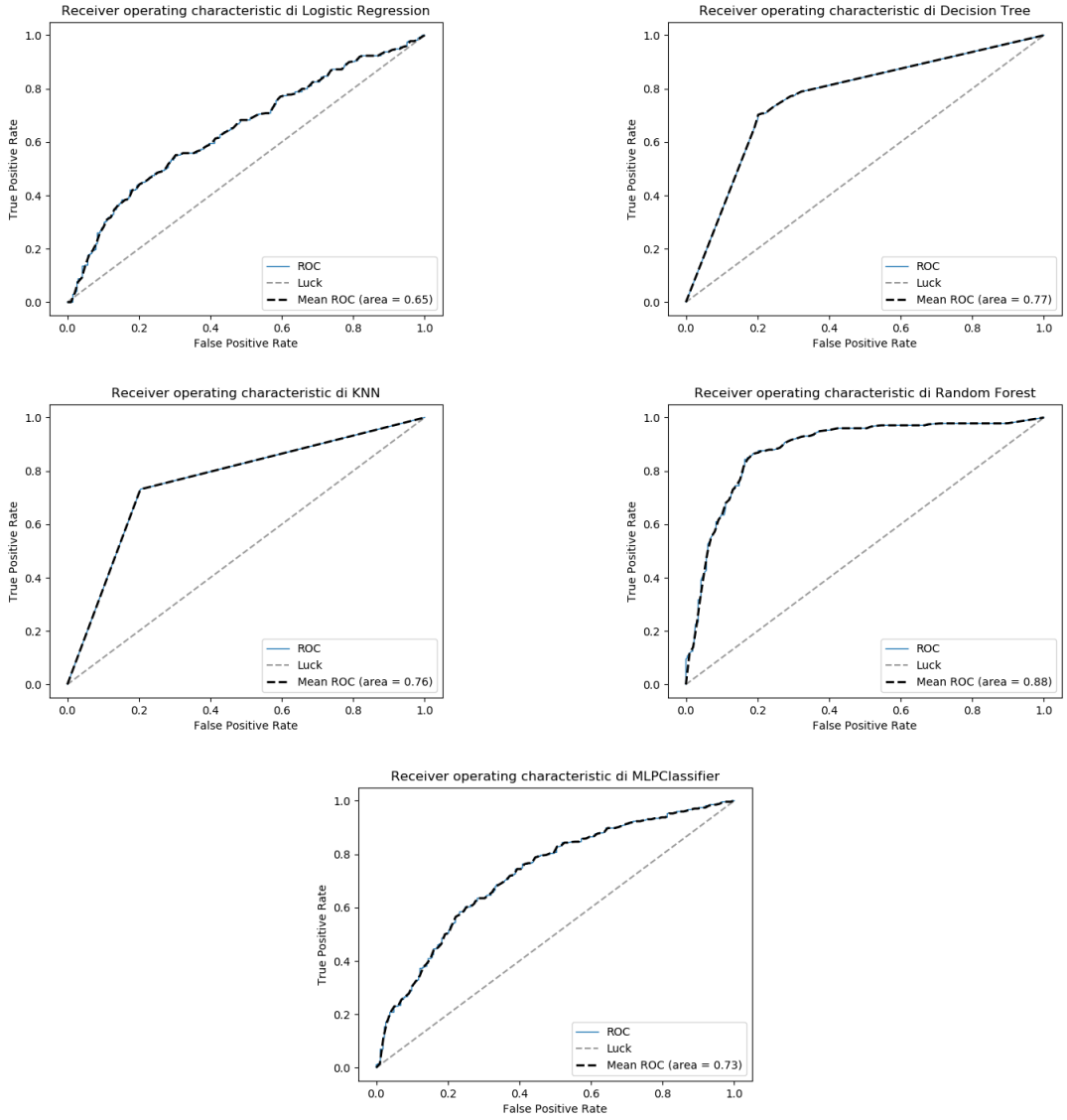


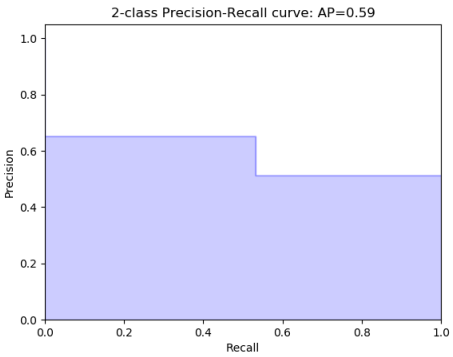
Figura 5.4: I risultati relativi alla metrica ROC per i cinque modelli analizzati

I grafici della ROC presentati in fig.5.4 confermano la bontà del modello Random Forest, che ottiene una AUC pari a 0.88 ed un andamento generale più simile ad un buon classificatore, e la inadeguatezza del modello di regressione logistica il cui grafico si avvicina a quello di un classificatore randomico. Invece, i modelli basati su K-Nearest Neighbors, Decision tree e rete neurale ottengono dei buoni risultati, con una AUC attorno allo 0.75, anche se non paragonabili al modello Random Forest.

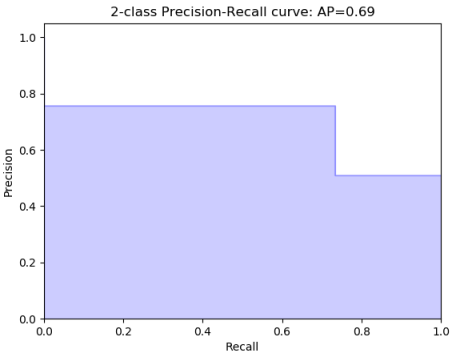
In fig.5.5 sono illustrati i grafici relativi alla curva precision-recall per tutti i metodi implementati in questo lavoro. Come ben visibile della figure il modello Random Forest ottiene dei buoni risultati (con una precision superiore all'80%) per valori di recall fino a 0.8. Tuttavia, per valori di richiamo più alti la precisione della predizione scende bruscamente fino al 50%, ossia la precisione di un classificatore randomico.

Per quanto riguarda i modelli di KNN (*K-Nearest Neighbors*) e Decision Tree si conferma l'idea che i risultati siano simili, sia considerando l'average precision, pari a 0.71 per gli alberi e 0.69 per KNN, sia per l'andamento della curva precision-recall che, in entrambi i casi, scende alla precisione del 50% con la recall al 0.7.

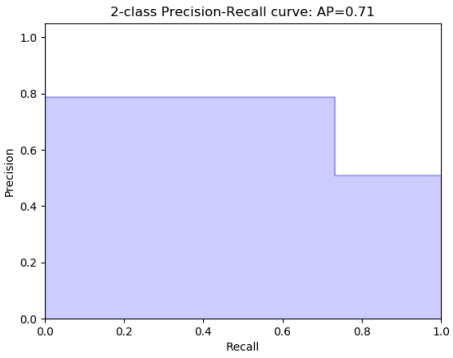
Infine i modelli di regressione logistica e rete neurale ottengono le prestazioni peggiori con una curva precision recall in cui la precisione scende già a partire da valori di recall pari a 0.6.



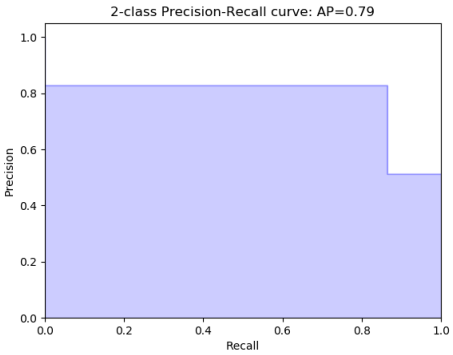
Regressione logistica



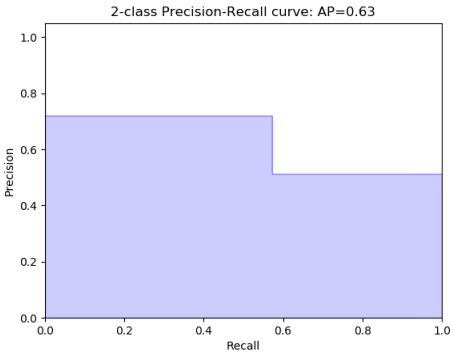
Decision tree



K-Nearest Neighbors



Random Forest



Rete neurale

Figura 5.5: I grafici Precision-Recall per i cinque modelli analizzati

5.4 Verifica delle ipotesi di lavoro

Per verificare le ipotesi esposte nel paragrafo 2.7 verrà quindi utilizzato il modello Random Forest in quanto è quello che ottiene i risultati migliori su tutte le metriche considerate. Si è dunque analizzata l'importanza delle feature nel modello ed è emerso che:

1. per la prima ipotesi dai dati ottenuti si evince che, nonostante il possesso di un diploma di laurea sia al 4 e 5 posto fra le feature importanti, il dottorato incide poco ai fini della predizione. L'ipotesi risulta dunque verificata a metà: se è vero che la presenza di un diploma di laurea risulta abbastanza importante al fine del successo della startup, un dottorato invece non influisce altrettanto;

2. il ranking universitario ha un peso molto basso sulla predizione, smentendo quindi l'ipotesi 2;

3. come è ben visibile nella tabella 5.2 le esperienze lavorative sono rispettivamente al primo (j_nYears) e al terzo (j_nPrevJobs) posto sulla scala di importanza delle feature. In particolare la durata della carriera lavorativa è di gran lunga la feature più importante, verificando quindi l'ipotesi 3;

4. l'eventuale esperienza in altre fondazioni di startup si colloca al secondo posto fra le feature importanti, ciò conferma l'ipotesi secondo cui sia una fra le caratteristiche necessarie al fine di una giusta valutazione del team.

Feature	**Weight**
j_nYears	0.32225485
j_serialEnt	0.09807538
j_nPrevJobs	0.09464689
ed_master	0.08565705
ed_bachelor	0.07590494
ed_other	0.0642577
j_moveToUSA	0.05670349
ed_ranking	0.05261391
ed_phd	0.05029912
ed_tech	0.04760573
ed_business	0.02062106

Tabella 5.2: La tabella riassuntiva dell'importanza delle varie feature nel modello Random Forest

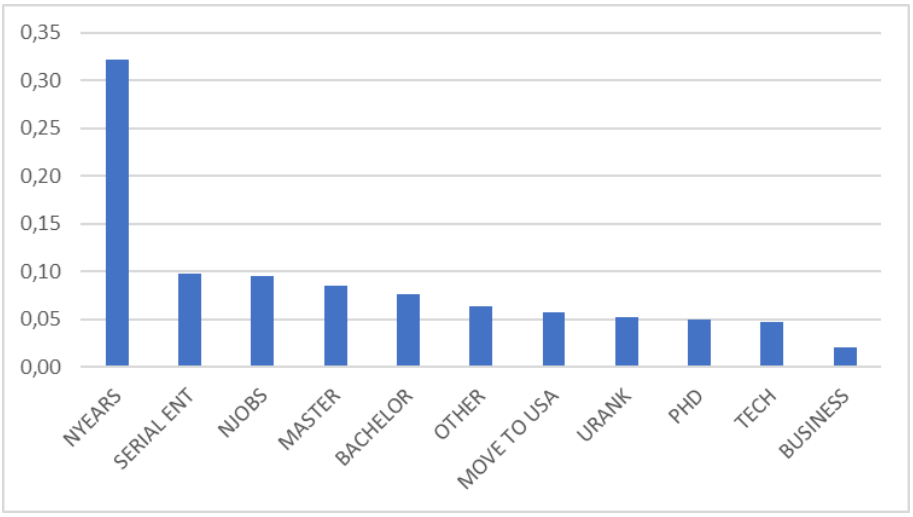


Figura 5.6: L'importanza delle feature nel modello Random Forest

Capitolo 6

Conclusioni

6.1 Limiti e sviluppi futuri

I principali limiti di questo lavoro, come già illustrato nel capitolo 3, riguardano la creazione del dataset e la struttura dello stesso.

Infatti, le limitazioni legate alle fonti di informazione riguardo le persone e, di conseguenza, alla standardizzazione dei dati sono importanti per la creazione di un dataset completo e accurato riguardo i founder delle startup.

Altro limite importante riguardante il dataset è la dimensione dello stesso: dagli originari 18.227 profili si è dovuto scendere, per limiti della piattaforma da cui si sono prelevati i dati, a 8.731 e, per quanto riguarda le imprese, si è operato un taglio netto considerando, al termine delle elaborazioni, solamente 3.586 startup. In tal senso si potrebbe procedere all'inclusione nel dataset delle imprese e dei profili mancanti al fine di aggiungere osservazioni ai modelli e, di conseguenza, migliorarli.

Gli sviluppi più importanti legati a questo lavoro includono, ad esempio, l'integrazione di ulteriori informazioni al dataset e il perfezionamento, seguendo le accortezze sopra riportate, dei dati ottenuti. In quest'ottica si può pensare di utilizzare sia tecniche di text mining, come ad esempio la NER (*Name Entity Recognition*), sia tecniche di term embeddings come *SkipGram* e *CBOW (Continuous Bag Of Words)*

In particolare si può pensare di includere fra le caratteristiche del team il settore in cui opera la startup e di incrociare i dati relativi ai campi di studio e all'ambito lavorativo dei soggetti con quest'ultimo per fornire un'indicazione più precisa sulla conoscenza specifica del team riguardo il mercato in cui si posiziona l'impresa.

6.2 Conclusioni

In questo lavoro si è cercato di realizzare uno studio dell'apporto del capitale umano nel processo di avvio di iniziative imprenditoriali, in particolare si è analizzato il peso delle singole caratteristiche di esperienza lavorativa e di formazione accademica del team dei founder tramite l'applicazione di tecniche di machine learning.

A tal fine si è proceduto alla creazione di un dataset custom basandosi sui dati presenti sul database Crunchbase ed integrandoli con le esperienze lavorative e professionali dei singoli soggetti.

Si è poi creato il profilo del team dei founder delle singole startup considerando le caratteristiche dei singoli e, attraverso una funzione di pesatura ad hoc, combinandole tenendo in considerazione le dinamiche di gruppo.

Per effettuare la predizione si sono analizzati cinque modelli di classificazione verificandone i risultati sulla base delle metriche di accuracy, precision, recall, ROC, AUC e F1.

Dei risultati ricavati il modello migliore è risultato quello ottenuto attraverso l'applicazione delle Random Forest, che è poi stato utilizzato per verificare le ipotesi di lavoro esposte nel par.2.7. Si è quindi arrivati a verificare che:

- l'esperienza lavorativa pregressa, intesa come durata della carriera, ha l'influenza maggiore sul successo della startup;

- il numero di posizione lavorative ha comunque un'influenza importante sul successo della startup;
- l'eventuale esperienza pregressa legata alla fondazione di startup influenza in maniera importante il successo dell'impresa;
- il grado di formazione accademica risulta importante solo per i gradi di istruzione universitaria ma non per i dottorati di ricerca;
- l'eventuale frequentazione di università prestigiose, come classificate da ARWU [24], non influisce sul successo della startup.

In definitiva la precisione del modello proposto risulta essere più che sufficiente per fornire un'indicazione sulla bontà dell'iniziativa imprenditoriale, fornendo un valido aiuto agli investitori pubblici e privati.

Bibliografia

[1] LinkedIn statistic overview page, *https://news.linkedin.com/about-us#statistics*, visitato il 24/07/2019.

[2] Eliasson, Gunnar. "The firm as a competent team." Journal of Economic Behavior Organization 13.3 (1990): 275-298.

[3] Nonaka, Ikujiro. "A dynamic theory of organizational knowledge creation." Organization science 5.1 (1994): 14-37.

[4] Muzyka, Dan, Sue Birley, and Benoit Leleux. "Trade-offs in the investment decisons of European venture capitalists." Journal of Business Venturing 11.4 (1996): 273-287.

[5] Vanaelst, Iris, et al. "Entrepreneurial team development in academic spinouts: An examination of team heterogeneity." Entrepreneurship Theory and Practice 30.2 (2006): 249-271.

[6] Clarysse, Bart, and Nathalie Moray. "A process study of entrepreneurial team formation: the case of a research-based spin-off." Journal of Business Venturing 19.1 (2004): 55-79.

[7] Baptista, Rui, Murat Karaöz, and Joana Mendonça. "The impact of human capital on the early success of necessity versus opportunity-based entrepreneurs." Small Business Economics 42.4 (2014): 831-847.

[8] Ratzinger, Daniel, et al. "The impact of digital start-up founders' higher education on reaching equity investment milestones." The Journal of Technology Transfer 43.3 (2018): 760-778.

[9] Fernando Muñoz-Bullon, Maria J. Sanchez-Bueno Antonio Vos-Saz (2015) Startup team contributions and new firm creation: the role of founding team experience, Entrepreneurship Regional Development, 27:1-2, 80-105

[10] Panel Study of Entrepreneurial Dynamics - University of Michigan `http://www.psed.isr.umich.edu/psed/home`

[11] Shalev-Shwartz, Shai, and Shai Ben-David. Understanding machine learning: From theory to algorithms. Cambridge university press, 2014.

[12] Hosmer Jr, David W., Stanley Lemeshow, and Rodney X. Sturdivant. Applied logistic regression. Vol. 398. John Wiley Sons, 2013.

[13] Paolo Medici, Elementi di Analisi per Visione Artificiale, 2017, `http://www.ce.unipr.it/people/medici/geometry.pdf`

[14] Breiman, Leo. Random forests. Machine learning, 2001, 45.1: 5-32..

[15] A.Pietracaprina - Big Data Computing - `http://www.dei.unipd.it/~capri/BDC/lectures.htm`

[16] Hu, Li-Yu Huang, Min-Wei Ke, Shih-Wen Tsai, Chih-Fong. (2016). The distance function effect on k-nearest neighbor classification for medical datasets. SpringerPlus. 5. 10.1186/s40064-016-2941-7.

[17] Cunningham, P., Delany, S. J. (2007). k-Nearest neighbour classifiers. Multiple Classifier Systems, 34(8), 1-17.

[18] Tay, Bunheang Hyun, Jung Keun Oh, Sejong. (2014). A Machine Learning Approach for Specification of Spinal Cord Injuries Using Fractional Anisotropy Values Obtained from Diffusion Tensor Images. Computational and mathematical methods in medicine. 2014. 276589. 10.1155/2014/276589.

[19] Rosenblatt, Frank. "The perceptron: a probabilistic model for information storage and organization in the brain." Psychological review 65.6 (1958): 386.

[20] Quiza, Ramon Davim, J. (2011). Computational Methods and Optimization. 10.1007/978-1-84996-450-0.

[21] Feed-forward and feedback networks - https://subscription.packtpub.com/book/big_data_and_business_intelligence/9781788397872/1/ch01lvl1sec21/feed-forward-and-feedback-networks, visitato il 17/08/2019

[22] Deep Learning CNN's in Tensorflow with GPUs - https://hackernoon.com/deep-learning-cnns-in-tensorflow-with-gpus-cba6efe0acc2, visitato il 17/08/2019

[23] Rokach, Lior, and Oded Maimon. "Decision trees." Data mining and knowledge discovery handbook. Springer, Boston, MA, 2005. 165-192.

[24] Academic Ranking of World Univesities - http://www.shanghairanking.com

[25] Clarivate Analytics - Highly Cited Researchers https://hcr.clarivate.com/

[26] Tom Quirk - LinkedIn API https://github.com/tomquirk/linkedin-api

[27] Wikipedia - Academic Degrees https://en.wiktionary.org/wiki/Appendix:Academic_degrees

[28] TechCrunch - Breaking a myth: Data shows you don't actually need a co-founder - https://techcrunch.com/2016/08/26/co-founders-optional/

[29] Spiegel, Olav, et al. "Going it all alone in web entrepreneurship?: a comparison of single founders vs. co-founders." Proceedings of the 2013 annual conference on Computers and people research. ACM, 2013.

[30] Chawla, Nitesh V., et al. "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research 16 (2002): 321-357.

[31] Scikit Learn - https://scikit-learn.org/

[32] Imbalanced learn - https://imbalanced-learn.readthedocs.io/en/stable/

[33] Breiman, Leo. "Some properties of splitting criteria." Machine Learning 24.1 (1996): 41-47.

[34] Kibriya, Ashraf M., and Eibe Frank. "An empirical comparison of exact nearest neighbour algorithms." European Conference on Principles of Data Mining and Knowledge Discovery. Springer, Berlin, Heidelberg, 2007.

[35] Flach, Peter. "The many faces of ROC analysis in machine learning." ICML Tutorial (2004).

[36] K-Fold and other cross validation techniques https://medium.com/datadriveninvestor/k-fold-and-other-cross-validation-techniques-6c03a2563f1e

Elenco delle figure