UNIVERSITÀ DI PARMA

Dipartimento di Ingegneria e Architettura
Corso di Laurea in Ingegneria Informatica, Elettronica e delle Telecomunicazioni

Tecniche di Machine Learning nell'analisi di preventivi di aziende grafico-editoriali

Machine Learning techniques applied to the analysis of graphic-publishing companies' quotes

Relatore:
Prof.ssa Monica Mordonini

Correlatore:
Prof. Michele Tomaiuolo

Tesi di Laurea di:
Vincenzo Fraello

ANNO ACCADEMICO 2020-2021

Ai miei genitori e mia sorella.

Indice

Introduzione

Questa tesi è suddivisa in sei capitoli. Il primo capitolo descrive lo stato dell'arte: cosa è stato già fatto sullo stesso argomento. In questa sezione si analizza il caso studio dal punto di vista economico (in quanto non è stato trovato alcun risultato che trattasse questo specifico argomento dal punto di vista scientifico).

L'elaborato prosegue, nel secondo capitolo, descrivendo il problema che si vuole risolvere e, gli approcci e i metodi con cui è stato risolto. Tale descrizione è stata esaminata ad un livello procedurale, al fine di poter introdurre il lettore all'argomento.

Dai capitoli tre a cinque si specifica nel dettaglio il lavoro di analisi e programmazione che è stato condotto.

Infine, nel sesto capitolo, si presentano i risultati che sono stati ottenuti, applicando i diversi approcci risolutivi al problema.

Nella sezione conclusioni si trovano delle ulteriori osservazioni, considerazioni finali e informazioni sugli sviluppi futuri del progetto.

Premessa

L'azienda *Logica s.r.l* è una software house che realizza sistemi informativi e presta consulenza organizzativa e gestionale, con una particolare specializzazione per l'Industria Poligrafica. Sviluppa avanzate soluzioni di tipo integrato, cioè sistemi ERP (Entreprise Resource Planning) o, con termine oggi più in voga nel settore, MIS (Management Information System), nonché soluzioni WEB based implementate in aziende poligrafiche di ogni tipologia produttiva e dimensione.[1]

Logica ha diverse tipologie di clienti, tutti in ambito grafico editoriale, ma con *struttura produttiva specifica* e di conseguenza con una tipologia di prodotto finito differente. Un prodotto stampato può variare dal libro, alla rivista, al packaging, alle etichette, ecc. In questo caso specifico, dopo attenta valutazione dei possibili database da analizzare, è stata scelta una base dati di un'azienda che ha internamente reparti e macchinari dedicati a prestampa, macchine offset 70x100 e 120x160, roto offset 16 e 48 pagine, finissaggio e legatoria (taglio, piega, fustellatura, vernice UV, punto metallico e brossura). La tipologia di prodotto che è stata analizzata in questo progetto sono gli *shopper* (*shopping bag* in carta con manici in corda) che vengono prodotti e venduti per il mercato europeo.

[1]Descrizione tratta dal sito web dell'azienda stessa. Home. (2021, March 31). Logica s.r.l. `https://www.logicasistemi.com`

Capitolo 1

Stato dell'arte

1.1 Un punto di vista economico

L'impiego di *Artificial Intelligence* (AI) e tecniche di *Machine Learning* (ML),[1] al fine di poter migliorare parti del processo produttivo e gestionale delle aziende, è una pratica molto diffusa oggi. Questi *sistemi di diagnostica intelligente* analizzano grandi quantità di dati, apprendono da essi e migliorano con l'esperienza, generando conoscenza e valore. Come si evince dalle parole di Hermann Simon e Martin Fassnacht [1]:

> "Artificial intelligence (AI) always generates considerable media excitement. The fundamental idea behind AI is simply to develop systems that automate analysis and decision-making processes normally carried out by human experts. One way to achieve this is through machine learning (ML), a set of algorithms that "learns" by itself, acquiring

[1] Il Machine Learning è un sottoinsieme dell'intelligenza artificiale (AI) che si occupa di creare sistemi che apprendono o migliorano le performance in base ai dati che utilizzano. Intelligenza artificiale è un termine generico e si riferisce a sistemi o macchine che imitano l'intelligenza umana. I termini apprendimento automatico e intelligenza artificiale vengono spesso utilizzati insieme e in modo interscambiabile, ma non hanno lo stesso significato. Un'importante distinzione è che sebbene tutto ciò che riguarda il machine learning rientra nell'intelligenza artificiale, l'intelligenza artificiale non include solo il machine learning. Machine learning—definito. (2021). Oracle Italia. `https://www.oracle.com/it/data-science/machine-learning/what-is-machine-learning`

> expert knowledge based on observed data, particularly historical data. Machine learning has been studied academically and applied in business for quite some time. Amazon's well-established recommendation engines are a classic example. Systems for up- and cross-selling no longer need to be continually reprogrammed by an expert; they update themselves by analyzing consumer behavior. The list of potential areas of application in pricing, marketing, and sales is impressive. An ML algorithm could be used to automate lead scoring, calculate price elasticity, predict customer choice, estimate willingness to pay, recommend a discount, predict churn rates, assess the win likelihood for a deal at a certain price, and identify the best targets for a promotion, to name just a few. While this sounds promising at first glance, companies must keep four major pitfalls in mind as they consider using ML:
>
> Applicability: Not all problems can be solved with machine learning. Machines can help solve problems that involve predicting a target variable, identifying patterns, classifying data items, or finding relationships. However, if the information that the machine needs to reach its conclusion cannot be observed in the data, ML will not provide any meaningful output. [...]"

l'AI è uno strumento che favorisce l'esecuzione e il completamento di processi decisionali rispettando i vincoli di efficacia ed efficienza. Infatti, se usata opportunamente, permette di ridurre notevolmente i tempi di completamento di *tasks aziendali* come, ad esempio, la realizzazione di un preventivo,[2] evitando così che il potenziale cliente possa rimanere in attesa indefinita. Occorre anche prestare attenzione al fatto che tali soluzioni non sono applicabili in qualunque caso. Gli algoritmi di ML non sono delle *black box* che forniscono sempre dei risultati corretti.

Sono numerosi i campi – e le attività di tali campi – in cui è possibile usufruire dei vantaggi che AI e ML forniscono. In questa tesi, si è deciso di focalizzare l'attenzione sul processo di preventivazione di prodotti di aziende del settore grafico-editoriale. In generale, il calcolo dei costi di un lavoro, che questo riguardi aziende grafiche, edili o di qualsiasi altra natura, è un'attività

[2] Che si fa in precedenza, [...] calcolo di previsione del costo di un determinato lavoro. In Treccani.it `https://www.treccani.it`

molto complessa. Pertanto, occorre avere piena conoscenza delle relative fasi di lavorazione dei prodotti e delle risorse disponibili, al fine di ottimizzare il più possibile le tempistiche, riducendo i tempi morti e rendendo al contempo il prototipo finale competitivo. Si tratta dunque di un *lavoro di programmazione*. AI e ML vengono incontro a queste esigenze, favorendo la realizzazione di moduli software intelligenti, in grado di offrire diverse tipologie di servizi. Esempi di questi ultimi possono essere: fornire un esito – caratterizzato da un certo intervallo di confidenza – circa la commissione del lavoro; gestire in autonomia le richieste di preventivi dei potenziali clienti distinguendole da altre; migliorare i software Configure, Price and Quote (CPQ)[3]. I tre esempi citati sebbene apparentemente differenti fra loro, in realtà, hanno un comune denominatore, ossia vengono sviluppati con l'intento di poter ottimizzare un processo aziendale, servendosi dei benefici offerti da AI e ML.

A supporto della validità del caso di studio trattato in questo elaborato, si riportano le parole di Louis Columbus [2]:

> "AI-based deal intelligence, pricing and predictive analytics are defining the future of CPQ selling today by providing real-time insights applicable to every sales cycle, from initial quote through contracts and renewals. AI and machine learning are making immediate contributions to driving more revenue by improving deal price guidance, deal intelligence, dynamic pricing and improving rebate & incentive management. Every CPQ vendor knows that pricing is the catalyst they need in their applications to attract and keep new customers."
>
> "improve close rates, improve quoting efficiency and improve subscription metrics."
>
> "Chief Revenue Officers (CROs), Sales and Sales Operations VPs are looking for CPQ solutions to step up and deliver improved pricing guidance across sales cycles"

[3] Configure, price quote (CPQ) software is a term used in the business-to-business (B2B) industry to describe software systems that help sellers quote complex and configurable products. Configure, price and quote. (2020, April 1). In Wikipedia. `https://en.wikipedia.org/wiki/Configure,_price_and_quote`

> "Pricing agility, intelligence and speed win more deals by enabling organizations to complete quotes faster and more completely than competitors."
>
> "Simplifying product, pricing, workflow and maintenance user experiences is enabling AI to make greater contributions to CPQ sales growth. Applying supervised and unsupervised machine learning to product, pricing and margin trade-offs is delivering solid revenue gains for those manufacturers using these techniques today based on conversations with CROs and Sales VPs."
>
> "CROs say that SAP CPQ is keeping up with how quickly their pricing, product configurations and workflows are changing while also providing customized views of data to several different groups of users, all leading to more closed deals."
>
> "Having a mobile CPQ app capable of auto-completing a configuration in seconds will increase quoting accuracy and improve sales close rates."
>
> "Machine learning needs to be the catalyst that enables a pricing ecosystem to keep improving by continually learning from transaction data. Target and stretch pricing can be improved using this technique"
>
> "Using the insights gained from AI and machine learning to provide workflow recommendations would save thousands of hours per year for Sales Operations teams."
>
> "CPQ selling strategies' effectiveness is improving thanks to AI and machine learning.".

Si noti l'importanza della frase con cui L. Columbus ha deciso di concludere l'articolo. Nella sua affermazione, viene messo in piena evidenza l'importante contributo di AI e ML nello specifico caso di strategie CPQ. Inoltre, nella sezione centrale del brano, fa un particolare riferimento ad un'azienda che offre soluzioni di questo tipo: *SAP SE* [3]. Tuttavia, queste tipologie di servizi vengono forniti anche da altri colossi come *Oracle*, *IBM* e *Salesforce* [4, 5, 6].

Tra le varie scelte possibili, si è deciso di riportare di seguito una definizione più dettagliata – ottenuta per via diretta dal sito di *Oracle* – di software CPQ [7]:

> "Oracle Configure, Price, Quote (CPQ) guides customers through a step-by-step process of entering a correctly configured and fully de-

> tailed purchase order. Once a correctly configured order is entered, the CPQ solution responds with a price quote including all the proper discounts, terms, and conditions for this specific customer. If the customer accepts the quote, it then becomes an order. The solution improves order accuracy, shortens sales cycles, and lowers operational costs. It is the only configure, price, quote solution available today that is designed to connect the front- and back-office."

Un altro particolare caso d'uso degno di nota è quello realizzato da *Automation Hero*: *detect intent in emails – Automate Price Quote Requests* [8]. L'intento di questa azienda è stato quello di realizzare un software intelligente, composto da vari moduli interagenti fra loro, in grado di ottimizzare la catena di operazioni – interpretazione delle richieste, analisi e definizione dei costi, invio di una risposta – che sono necessarie per la conclusione di un affare. Il cliente finale che utilizza il software è un'azienda di logistica che riceve richieste di preventivi tramite diversi canali, ad esempio, le e-mail.

In un articolo l'azienda mette in evidenza il fatto che rispetto alla vecchia gestione manuale, il nuovo sistema – basato sull'uso di AI e ML – garantisce tempi di risposta fino a tre volte più veloci, lasciando i clienti maggiormente soddisfatti. Di seguito si riporta un estratto che dettaglia il funzionamento del sistema:

> "A lot of time and effort is spent on these types of customer requests with low productivity, resulting in long waiting cycles for customers up to the point that sometimes the request is never even answered. Not only is the customer experience negative, but potential business opportunities are missed. It was determined that intelligent data recognition (IDR) technology should be utilized to process unstructured content from email requests."
>
> "Our experts leveraged the power of artificial intelligence (AI) in Hero_Flow and created two AI models to solve the problem. The first analyzed incoming emails to find out the intent and automatically selected the ones that were identified as quote requests. After filtering out the quote requests, the second model analyzed the unstructured text of each email to find out the quantity and weight of the packages the user wanted to ship, the origin and destination and the requested date of shipment. Once this information had been extracted, Hero_Flow looked up the price in another system and then composed

> and sent a reply email to the customer to inform them about the availability and cost of their planned transaction. The entire labor-intensive and error-prone process was replaced by an automated intelligent data recognition workflow, which involved no manual intervention. The result was a fast response time and a much more satisfying customer experience.".

In conclusione di questo capitolo, si può osservare come pur intervenendo in maniera differente, l'obiettivo da ottenere e i processi, nonché gli strumenti che consentono di raggiungerlo sono gli stessi nei tre casi appena citati. Fornendo numerosi esempi di preventivi accettati e non accettati o, configurazioni ottimali e non ottimali o, ancora, richieste e non richieste di preventivi, si addestra un sistema informatico – il quale produrrà un modello che userà per effettuare le predizioni – con lo scopo di risolvere problemi più o meno complessi.

Capitolo 2

Metodologie e approcci al problema

2.1 Analisi di preventivi

Proto è un ERP (Enterprise resource planning) per il settore delle aziende grafico editoriali composto da moduli software dove ciascun modulo gestisce i diversi processi produttivi e gestionali del settore. In particolare, il processo che andremo a prendere in esame in questo progetto è quello della preventivazione; ovvero, della definizione tecnico-economica di ciascun prodotto in termini di prototipazione per arrivare, dalle specifiche tecniche, alla definizione di un possibile prezzo di vendita. Il modulo software in oggetto è pertanto Proto-PREV.

Ciascun prodotto del settore grafico (es. libro, rivista, scatola, etichetta, depliant, cartello vetrina) si compone di una serie di parti/componenti che vengono prodotti ed elaborati attraverso i diversi processi produttivi di prestampa, stampa e allestimento e sono composti di materiali (es. carta, cartone, inchiostri) lavorati attraverso processi interni e/o esterni (conto lavoro).

Proto-PREV consente pertanto di definire quanto sopra attraverso un'interfaccia utente (Figura 2.1) che si compone di schede base guidate da una

struttura ad albero come sottoindicato:

- Testata – ove vengono inseriti i dati generali del lavoro per il quale si realizza il preventivo (es. cliente, agente, data di consegna, descrizione, codici identificativi di gare d'appalto, codici di richiesta cliente);
- Prodotti – ove vengono inseriti i dati di dettaglio del prodotto finito (es. tipo prodotto, tiratura, formato finito) e le lavorazioni di allestimento, imballo e trasporto;
- Parti – ovvero le componenti del prodotto (es. pagine, copertina, sovracoperta) – i cui dati produttivi sono il tipo di supporto di stampa (es. carta, tela, cartone), le sue caratteristiche (grammatura, tipologia, spessore ecc.) i colori di stampa della singola parte;
- Impianti – ovvero le messe in macchina in base alla resa sul foglio. In questo ambito vengono definiti i processi produttivi di dettaglio, ovvero la stampa con tutte le sue caratteristiche tecniche, la macchina, il formato di stampa, le lavorazioni speciali (es. plastifica, fustellatura, cordonatura, verniciatura);
- Capitolato – che riepiloga i dati produttivi, le quantità e i costi di produzione;
- Riepilogo – che riassume i margini di produzione per la definizione del prezzo finale di vendita.

Ogni preventivo può essere richiesto dal cliente in diverse *varianti*, che possono differire tra loro in relazione al numero di copie richiesto (in gergo tirature) o sulla base delle specifiche tecniche del prodotto stesso (es. numero di pagine di una rivista). Per ogni preventivo, e relative varianti, sulla base dei costi di produzione e dei margini di vendita attesi, verranno elaborate delle offerte al cliente. Se il cliente accetta una di queste offerte, la sola variante accettata diventa ordine di produzione. Ovviamente, non tutti preventivi elaborati diventano effettivo ordine di vendita, in quanto il cliente potrebbe

non accettare la proposta. L'obiettivo che si vuole raggiungere per mezzo del sistema di diagnostica intelligente è:

i. *Fornire una previsione con un certo intervallo di confidenza circa la commissione del lavoro in relazione al preventivo realizzato, analizzando uno storico passato dei vari preventivi fiscali (caratterizzati da certi parametri)*;

ii. *Fornire al produttore delle possibili informazioni sulle variabili che rendono competitivo il suo preventivo per fare eventuali scelte tecnologiche/organizzative che lo portino a colmare eventuali gap ed ottimizzare la produzione, aumentando la possibilità di acquisire lavori.*

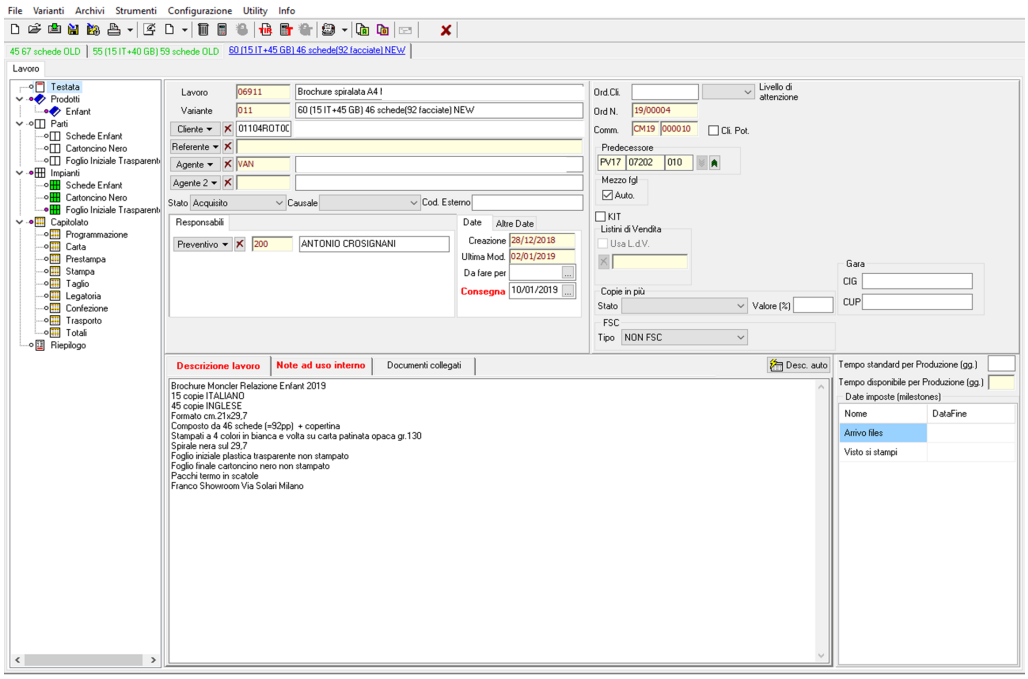


Figura 2.1: Schermata dell'interfaccia utente del software *Proto-PREV*

2.2 Machine Learning

Quando si parla di apprendimento automatico – in lingua inglese *Machine Learning* – si intende la capacità di un algoritmo di: (i) prendere delle decisioni in relazione a quanto appreso da una storia passata; (ii) migliorare con l'esperienza. Una definizione più rigorosa è stata data dall'informatico *Tom Michael Mitchell* [9]:

> "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Esistono tre tipologie di ML:

- *Apprendimento supervisionato*;
- *Apprendimento non supervisionato*;
- *Apprendimento semi-supervisionato.*

In tutti e tre i casi occorre fornire alla macchina una serie di esempi da analizzare. Una volta identificati i *pattern* tra i dati – ovvero le regolarità – questa genera un modello da utilizzare per risolvere il problema di partenza; nello specifico caso di questa tesi, quello presentato all'inizio di questo capitolo (2.1).

2.2.1 Apprendimento supervisionato

Uno degli approcci scelti per affrontare e risolvere il problema in oggetto è quello dell'apprendimento supervisionato. Tale tecnica di apprendimento automatico utilizza un insieme di dati che viene suddiviso in *training-set* e *test-set*. Ogni istanza dell'insieme dei dati di allenamento rappresenta una singola osservazione caratterizzata da un'etichetta, che la colloca nella relativa classe di appartenenza. Gli algoritmi – istruiti con i dati etichettati – producono un modello che viene usato per effettuare le predizioni. Viceversa, l'insieme dei dati di test non è dotato di alcuna etichetta e, viene utilizzato

come insieme di confronto per valutare la bontà del modello prodotto dall'algoritmo di ML. La Figura 2.2 mostra tramite un diagramma di flusso il processo di apprendimento supervisionato.

Nel caso del sistema di diagnostica intelligente, si è avuta una fase preliminare di cooperazione con gli esperti del dominio. L'obiettivo di questa fase è la raccolta dei dati dal database dell'azienda con un alto contenuto informativo circa il prodotto. Ogni informazione raccolta rappresenta una caratteristica del preventivo. In seguito, sono state esplorate le variabili raccolte tramite tecniche di analisi univariata e bivariata, al fine di poter comprendere la natura dei dati e, avere un'idea del significato dei risultati prodotti dagli algoritmi.

Nello step successivo, è avvenuta la manipolazione dei dati con lo scopo di rendere il dataset elaborabile dagli algoritmi di ML. Ne è un esempio *One-Hot Encoding*. Si sono poi addestrati gli algoritmi e valutati i modelli prodotti con l'ausilio di diverse metriche: *accuracy*, *precision*, *recall*, *confusion matrix* e *AUROC*. Il passo finale è stato quello di riduzione delle dimensionalità delle caratteristiche del preventivo e successivo ri-addestramento degli algoritmi sul dataset ridotto, con conseguente valutazione del modello tramite le metriche citate sopra.

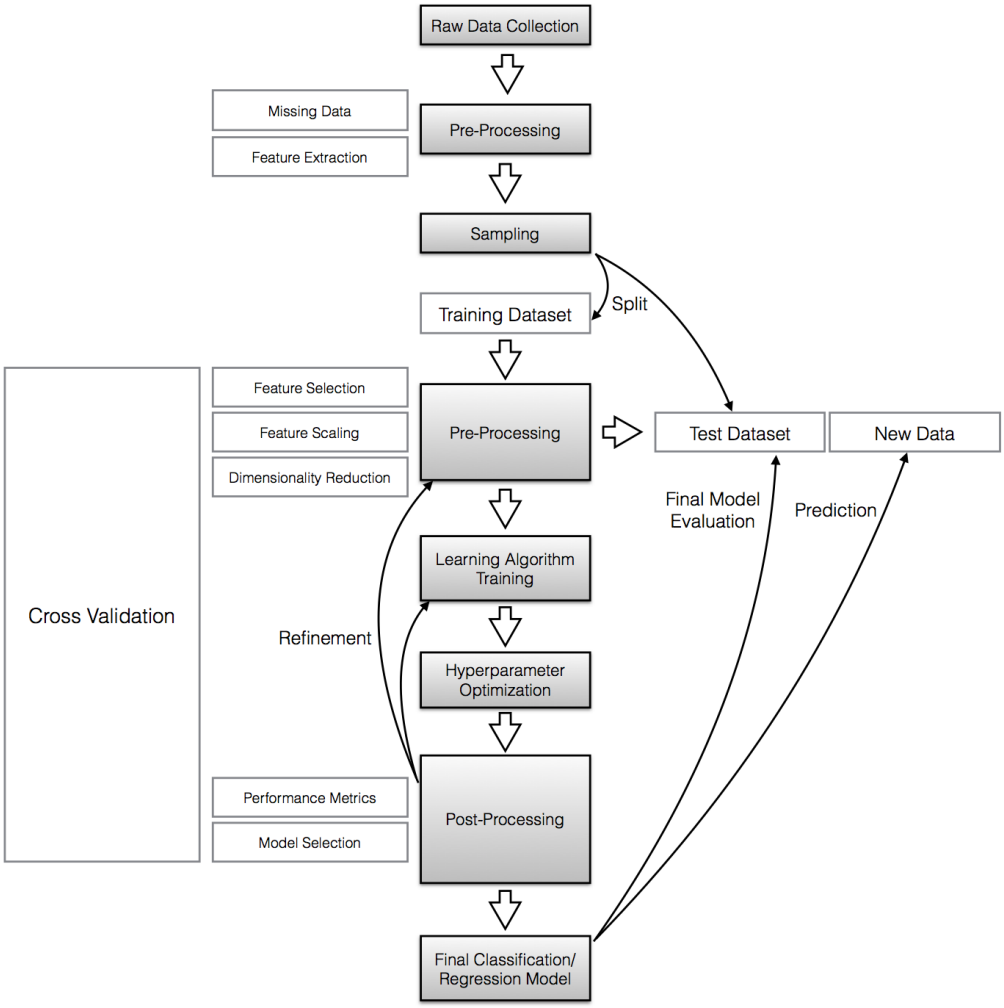


Figura 2.2: Diagramma di flusso dell'*apprendimento automatico supervisionato*

2.2.2 Apprendimento non supervisionato

Dopo aver analizzato i preventivi con algoritmi di apprendimento automatico supervisionato, si è deciso di perseguire un'altra strada: apprendimento non supervisionato. A differenza della prima metodologia, gli algoritmi non ricevono dati etichettati. Il sistema informatico stesso deve analizzare i dati, trovare le caratteristiche comuni e creare dei gruppi – in gergo *clusters* –

corrispondenti alle classi. La Figura 2.3 mostra lo schema di apprendimento non supervisionato.

Dopo aver applicato tali tecniche di analisi, si sono avuti risultati migliori – in termini di proporzioni dei clusters – con i dati sottoposti ad un processo di *scaling*: la standardizzazione. Tuttavia, non avendo dei dati di test, non è stato possibile valutare la bontà del modello prodotto dall'*algoritmo di clustering*.

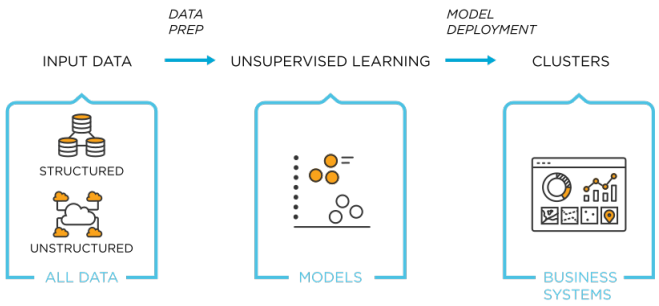


Figura 2.3: Schema *apprendimento automatico non supervisionato*

2.2.3 Apprendimento semi-supervisionato

Questa metodologia di apprendimento automatico combina le tecniche descritte nelle due sezioni precedenti di questo capitolo (2.2.1 e 2.2.2).

In questo caso si fornisce al sistema un insieme di dati di cui, solo una minoranza risulta essere etichetta. La restante parte maggioritaria non possiede un'etichetta. Gli algoritmi hanno il compito di creare dei clusters in relazione ai *pattern* tra i dati e, assegnare a tutti gli elementi di un cluster, l'etichetta degli esempi *labeled* al suo interno. Questa tecnica risulta particolarmente utile quando si ha una grande mole di dati non etichettati. Infatti, etichettandone solamente una parte, si riesce comunque ad ottenere un risultato risparmiando molto tempo. Nel caso in esame di questa tesi non è stato necessario sfruttare tale metodologia di apprendimento automatico, in quanto tutti i preventivi possiedono l'etichetta: "Approvato" - "Non approvato".

Capitolo 3

Strumenti utilizzati

3.1 Python

Python è un linguaggio di programmazione di alto livello, orientato agli oggetti, caratterizzato dalla presenza di una vasta gamma di librerie e moduli software che lo rendono molto versatile. Infatti, sono numerosi i campi di applicazione di questo linguaggio [10]: (i) *Web and Internet Development*; (ii) *Scientific and Numeric computing*; (iii) *Education*; (iv) *Software Development*; (v) *Business Applications* (ad esempio sistemi ERP).

Tra gli altri usi, si presta bene nell'ambito di analisi di grandi quantità di dati, intelligenza artificiale e apprendimento automatico. Le ragioni vengono esposte in un articolo della società *Towards Data Science*, pubblicato sulla piattaforma *Medium* [11]:

- "A great library ecosystem": esistono alcune librerie che forniscono algoritmi di ML e altre per l'elaborazione e visualizzazione dati;
- "A low entry barrier": lo sforzo richiesto per apprendere il linguaggio è limitato. Sin da subito, si è in grado di programmare, in quanto i comandi riprendono l'inglese comunemente parlato;
- "Flexibility": permette la combinazione di diversi stili e linguaggi di programmazione;

- "Platform independence": può essere eseguito su qualsiasi piattaforma. Alcuni esempi sono: *Windows*, *MacOS*, *Linux*, *Unix*;
- "Readability": essendo ispirato all'inglese comune, il processo di comprensione di codice scritto da qualcun altro viene facilitato;
- "Good visualization options": nel campo della *data science* una delle fasi più importanti è "presentare e comunicare i risultati". Python fornisce diverse librerie per la creazione di grafici;
- "Community support": la presenza di una *community* di supporto favorisce la risoluzione di problematiche in tempi molto brevi;
- "Growing popularity": in virtù delle motivazioni sopraelencate, Python è un linguaggio molto popolare, diffuso e richiesto.

3.1.1 NumPy

Nel linguaggio di programmazione Python tutto è un oggetto. Di conseguenza, il suddetto linguaggio, si presta poco al *number-crunching*[1], in quanto la memoria viene utilizzata in maniera poco efficiente[2]. Tuttavia, il problema si aggira sfruttando la capacità di Python di lavorare bene anche con altri linguaggi di programmazione, come ad esempio *C* o *Fortran*, i quali utilizzano la memoria in maniera più ottimizzata. *NumPy* è una libreria *open source* per il calcolo scientifico, sviluppata per mezzo di linguaggi di programmazione predisposti all'elaborazione di grandi quantità di dati e in grado di effettuare calcoli complessi in tempi molto brevi [12, 13].

Questa libreria fornisce supporto agli *array n-dimensionali*. Internamente gli array n-dimensionali sono caratterizzati dalla presenza di due elementi:

[1] Mathematical work performed by people or computers that involves large amounts of information or numbers. Number-crunching. Cambridge Dictionary. (2021, April 28). `https://dictionary.cambridge.org/it/dizionario/inglese/number-crunching`

[2] *Python* risulta essere più lento di almeno un fattore 200 rispetto a *C++*.

- *Data buffer* – ovvero blocchi di memoria contigui, come nel caso di array C o Fortran;
- *Metadata* – ovvero le informazioni aggiuntive sui dati del *data buffer* (es. *offset*, *stride*, *dtype*).

Quando si effettuano operazioni sui dati, come ad esempio il *reshape*, non si crea un nuovo array, bensì, dei metadati con parametri differenti che fanno riferimento allo stesso *data buffer*.

3.1.2 Pandas

Pandas è una libreria basata su *NumPy* che offre strutture dati e strumenti per l'analisi e la manipolazione dei *Big Data*. Esempi di operazioni eseguibili con questa libreria sono: importazione dataset, pulizia dataset, normalizzazione o standardizzazione dataset, visualizzazione dataset [14, 15].

3.1.3 Matplotlib

L'ultimo dei cinque passi fondamentali della *data science* è la fase in cui si devono comunicare e presentare i risultati, con lo scopo di spiegarli in maniera chiara. Il modo migliore di farlo è sicuramente attraverso i grafici. Questa libreria – tramite l'aiuto di librerie come *Pandas* e *Numpy* – permette la creazione di diverse tipologie di grafici: istogrammi, a linee, a barre, a scatola e baffi, a dispersione [16].

3.1.4 Seaborn

Seaborn è una libreria di Python basata su *Matplotlib* che con l'ausilio di altre librerie, come *Pandas* e *Numpy*, permette di tracciare grafici per la visualizzazione dei dati. Può essere considerato come un *superset* della libreria Matplotlib [17, 18].

3.1.5 Scikit-learn

Scikit-learn è una libreria *open source* – basata sull'uso di *NumPy*, *SciPy* e *Matplotlib* – contenente una vasta gamma di strumenti utili per l'apprendimento automatico supervisionato e non supervisionato. Contiene algoritmi per la classificazione, regressione, *clustering*, riduzione delle dimensionalità, selezione dei modelli, *tuning* dei parametri e *preprocessing* dei dati [19].

3.2 Jupyter

Jupyter Notebook è un *framework* usato per creare frammenti di codice eseguibili in maniera indipendente con vari linguaggi di programmazione come ad esempio: *Python*, *Julia*, *Scala*, *R*, *C++* [20].

3.3 Structured Query Language

SQL è l'acronimo per *Structured Query Language*. SQL è un linguaggio usato per: (i) creazione e modifica di schemi di databases relazionali; (ii) inserimento e gestione di valori nella base di dati; (iii) interrogazione delle basi di dati per ottenere delle informazioni da queste [21].

Figura 3.1: Loghi degli strumenti software utilizzati

Capitolo 4

Recupero dei dati e costruzione del dataset

4.1 Analisi tabelle della base di dati

La scienza dei dati, come mostrato in Figura 4.1, è frutto della combinazione di discipline appartenenti, almeno in un caso, a settori differenti: (i) matematica e statistica; (ii) informatica; (iii) esperienza o conoscenza del dominio applicativo.

Durante lo sviluppo del progetto in esame, matematica e statistica sono state applicate in maniera indiretta attraverso gli algoritmi di *machine learning*. Diversamente, esperienza del dominio e informatica, sono state sfruttate per via diretta, al fine di poter individuare ed estrarre dalla base di dati le *caratteristiche* utili dei preventivi. Infatti, nella fase iniziale del progetto, si sono tenute delle riunioni in collaborazione con gli esperti del dominio, i quali hanno illustrato da un lato il processo produttivo di un prodotto grafico applicato al software *Proto-PREV* e, dall'altro, la struttura interna del database relazionale. La prima parte – ovvero la comprensione degli elementi che compongono un preventivo – è stata propedeutica per la seconda, in quanto ad ogni componente del preventivo corrisponde una tabella nella base di dati.

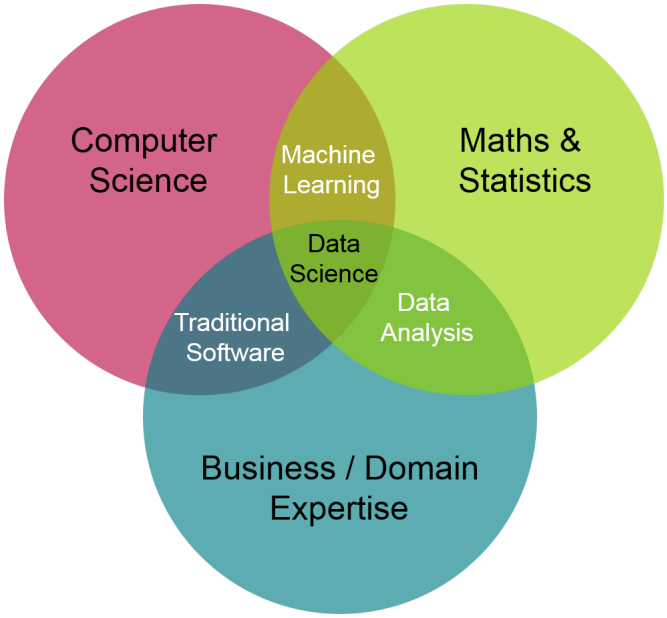


Figura 4.1: Diagramma di Venn della *data science*

Logica, l'azienda con cui è avvenuta la collaborazione, ha fornito dei dati reali (anonimizzati al fine di poter garantire la *privacy* dei loro clienti) di preventivi dei prodotti *shopping bag in carta con manici in corda* a partire dal loro database interno. Durante le riunioni con gli esperti del dominio, sono stati discussi i concetti generali dei preventivi di prodotti di aziende grafico-editoriali e di come questi vengano memorizzati nel database: (i) ogni prodotto grafico prende il nome di JOB; (ii) ogni JOB è caratterizzato da una stringa univoca di diciotto caratteri che prende il nome di JobID; (iii) ogni prodotto può avere delle varianti; (iv) ogni variante è un JOB a sé stante; (v) più varianti di uno stesso prodotto hanno stessa testata, ma potrebbero avere parti, impianti e riepiloghi differenti.

Nella sezione 2.1 del capitolo 2 è stata presentata la struttura di un preventivo generico di un prodotto grafico-editoriale. Le corrispondenze tra tabella della base di dati e, componente del preventivo, sono qui sottoelencate:

- JOBHDR (Testata);

- JOBPROD (Prodotti);
- JOBPART (Parti);
- JOBIMP (Impianti);
- JOBROW (Capitolato).

Di tali tabelle, solo alcune proprietà sono state prese in considerazione. Di seguito, sono riportati i campi delle tabelle analizzate:

JOBHDR: in questa tabella sono contenute le testate dei preventivi.

- JobID: stringa di 18 caratteri che identifica univocamente un JOB;
- CodReg: numero del registro a cui appartiene il preventivo (annata del preventivo es. pv20);
- NumReg: numero progressivo del preventivo appartenente ad un registro (ci sono clienti che vogliono che il numero riparta da 1 ad ogni nuovo registro e altri che vogliono un numero progressivo infinito);
- CodVar: codice progressivo che indica il numero della variante;
- Approved: *flag* che indica quale è la variante approvata (Variante approvata = 1, Variante non approvata = 0);
- NumOrd: tale campo indica che il preventivo è stato accettato;
- CMNUMREG: tale campo indica che il preventivo è stato accettato ed è andato in produzione.

JOBPROD: in questa tabella sono contenuti i dati di dettaglio dei prodotti finiti.

- JobID: stringa di 18 caratteri che identifica univocamente un JOB. Il prodotto ha lo stesso JobID della testata del preventivo perché appartengono allo stesso JOB;

- CodReg: numero del registro a cui appartiene il preventivo (annata del preventivo es: pv20). Il prodotto ha lo stesso CodReg della testata del preventivo perché appartengono allo stesso JOB;
- NumReg: numero progressivo del preventivo appartenente ad un registro (ci sono clienti che vogliono che il numero riparta da 1 ad ogni nuovo registro e altri che vogliono un numero progressivo infinito). Il prodotto ha lo stesso NumReg della testata del preventivo perché appartengono allo stesso JOB;
- ProdID: codice progressivo identificativo del prodotto;
- IDx: posizione del prodotto nella lista grafica del software Proto-PREV;
- Codice: identifica il tipo del prodotto (categorizzazione dei prodotti: riferimento a Codice della tabella TPPROD);
- Nome: stringa arbitraria editabile;
- FmPag: formato finale del prodotto;
- Copie: numero di copie del prodotto finito (in gergo tiratura);
- LivDif: livello di difficoltà nella realizzazione del prodotto (Facile = 0, Medio = 1, Difficile = 2). Al crescere del livello di difficoltà di realizzazione del prodotto, cresce il tempo necessario per la sua produzione e, di conseguenza, i costi.

TPPROD: tabella che contiene i prodotti.

- Codice: codice del prodotto;
- Nome: nome del prodotto.

JOBPART: in questa tabella sono contenute le componenti dei prodotti. Le parti sono da considerarsi come delle "forme" per la stampa.

- JobID: stringa di 18 caratteri che identifica univocamente un JOB. La parte del prodotto ha lo stesso JobID della testata del preventivo perché appartengono allo stesso JOB;
- CodReg: numero del registro a cui appartiene il preventivo (annata del preventivo es: pv20). La parte del prodotto ha lo stesso CodReg della testata del preventivo perché appartengono allo stesso JOB;
- NumReg: numero progressivo del preventivo appartenente ad un registro (ci sono clienti che vogliono che il numero riparta da 1 ad ogni nuovo registro e altri che vogliono un numero progressivo infinito). La parte del prodotto ha lo stesso NumReg della testata del preventivo perché appartengono allo stesso JOB;
- PartID: codice progressivo identificativo della parte;
- IDx: posizione della parte nella lista grafica del software Proto-PREV;
- Codice: identifica il tipo di parte (equivalente del codice prodotto);
- Nome: stringa arbitraria editabile;
- Num: numero di parti utilizzate per realizzare il prodotto finito (es. se il prodotto ha 32 pagine e lo sviluppo avviene in segnature di tipo 16mi, serviranno due segnature 16mi);
- NumPag: numero di pagine che si producono con una parte (es. sedicesimi);
- Formato aperto della parte (per formato aperto si intende l'esploso del formato chiuso in base al singolo componente che si analizza, es. se un libro è 15 x 20 formato chiuso, il formato della copertina sarà base x 2 + lo spessore del dorso (es. 1.8 cm) x altezza = [(15 x 2)+1.8] x 20 = 31.8 x 20);
- ColBi: numero di colori in bianca;

- ColVo: numero di colori in volta;
- PcCopBi: coprenza in bianca. Indica la percentuale di inchiostro usato in bianca;
- PcCopVo: coprenza in volta. Indica la percentuale di inchiostro usato in volta;
- CodCar: campo che contiene il codice della carta selezionata. Il codice si riferisce all'archivio delle carte;
- GrCar: grammatura della carta.

CARTA:[1] tabella che contiene le carte di listino.

- Codice: codice univoco identificativo della carta. Fornisce la qualità della carta;
- Nome: nome della carta.

JOBIMP: in questa tabella sono contenuti i dati di dettaglio dei processi produttivi.

- JobID: stringa di 18 caratteri che identifica univocamente un JOB. L'impianto ha lo stesso JobID della testata del preventivo perché appartengono allo stesso JOB;
- CodReg: numero del registro a cui appartiene il preventivo (annata del preventivo es: pv20). L'impianto ha lo stesso CodReg della testata del preventivo perché appartengono allo stesso JOB;

[1] Quando viene elaborata una stima di prezzo per un prodotto si possono fare due scelte: (i) utilizzare un listino di carta con ipotesi di prezzo "a prossimo riacquisto". Considerando che la carta è la materia prima del prodotto, ha un'incisione del 25-30% sul prezzo finale e un prezzo sul mercato che varia continuamente; (ii) utilizzare il prezzo della carta giacente a magazzino, con l'idea che sia disponibile in caso di acquisizione dell'ordine. Nel caso del progetto in analisi, si è deciso di utilizzare il listino di carta con ipotesi di prezzo.

- NumReg: numero progressivo del preventivo appartenente ad un registro (ci sono clienti che vogliono che il numero riparta da 1 ad ogni nuovo registro e altri che vogliono un numero progressivo infinito). L'impianto ha lo stesso NumReg della testata del preventivo perché appartengono allo stesso JOB;
- ImpID: codice progressivo identificativo dell'impianto;
- IDx: posizione dell'impianto nella lista grafica del software Proto-PREV;
- Quantità: indica il numero di impianti (messe in macchina) che si devono fare;
- TipoImp: indica il tipo di impianto usato (Nessuno = 0, Bobina = 1, Modulo = 2, Digitale = 3, Multi-bobina = 4, Semi-rotativa = 5, Cil. Interc. = 6, Digitale bobina = 7, Digitale foglio = 8);
- FmCar: Formato della carta usata;
- CsCar: costo della carta al chilogrammo;
- BV (Bianca e volta): *flag* che, nel caso di fogli stampati su due lati, indica se la macchina è in grado di girare in autonomia il foglio oppure se occorre l'ausilio dell'operatore.

JOBPASS: in questa tabella sono contenuti i dettagli delle macchine da stampa, legatoria e di altre macchine del settore. Per ogni JOBIMP si hanno uno o più JOBPASS.

- JobID: stringa di 18 caratteri che identifica univocamente un JOB. La passata ha lo stesso JobID della testata del preventivo perché appartengono allo stesso JOB;
- CodReg: numero del registro a cui appartiene il preventivo (annata del preventivo es: pv20). La passata ha lo stesso CodReg della testata del preventivo perché appartengono allo stesso JOB;

- NumReg: numero progressivo del preventivo appartenente ad un registro (ci sono clienti che vogliono che il numero riparta da 1 ad ogni nuovo registro e altri che vogliono un numero progressivo infinito). La passata ha lo stesso NumReg della testata del preventivo perché appartengono allo stesso JOB;
- ImpID: codice progressivo identificativo dell'impianto;
- PassID: codice progressivo identificativo della passata;
- CodMac: codice della macchina (riferimento all'archivio macchine);
- NumCl: numero di cambi lingua. Il numero di lingue è dato da NumCl+1;
- TempoAvv1: indica il tempo (in decimi di ore) che intercorre dalla fase di avviamento alla fase di regime della macchina. Nella fase di avviamento, i fogli prodotti vengono buttati (scarti di avviamento);
- ScAvv: indica il numero di fogli scartati durante la fase di avviamento;
- ScTir: scarti di tiratura;
- FgFin: fogli finiti;
- FgTir: fogli di tiratura (FgFin + ScTir);
- FgDati: fogli totali dati alla macchina (FgTir + ScAvv).

ARCHIVIO MACCHINE: in questa tabella sono contenute le informazioni sui macchinari dell'azienda.

- Codice: identificativo della macchina.

JOBROW: in questa tabella sono contenute tutte le lavorazioni necessarie per realizzare il prodotto.

- JobID: stringa di 18 caratteri che identifica univocamente un JOB. La lavorazione del prodotto ha lo stesso JobID della testata del preventivo perché appartengono allo stesso JOB;
- CodReg: numero del registro a cui appartiene il preventivo (annata del preventivo es: pv20). La lavorazione del prodotto ha lo stesso CodReg della testata del preventivo perché appartengono allo stesso JOB;
- NumReg: numero progressivo del preventivo appartenente ad un registro (ci sono clienti che vogliono che il numero riparta da 1 ad ogni nuovo registro e altri che vogliono un numero progressivo infinito). La lavorazione del prodotto ha lo stesso NumReg della testata del preventivo perché appartengono allo stesso JOB;
- Enabled: *flag* che indica se si effettua la lavorazione sul prodotto (Lavorazione attiva = 1, Lavorazione non attiva = 0). Si considerano solo le lavorazioni attive;
- EnabFlag: codice della lavorazione;
- CodNuc: codice del nucleo. Il nucleo è l'unità produttiva. (es. nucleo di stampa, di legatoria, di piega, di lavorazioni manuali). In genere, c'è un nucleo per macchina da stampa, ma un nucleo di stampa può avere più macchine da stampa associate. Una macchina da stampa può essere attrezzata/configurata in maniera differente dalla versione di base;
- IEM: lavorazione Interna, lavorazione Esterna (fornitore esterno), lavorazione materiale (es. carta, inchiostro);
- RigaID: indica l'elemento a cui si riferisce una lavorazione;
- ASSRighe: la lavorazione può essere di prodotto, parte o impianto. Questo campo indica il tipo di lavorazione (Prodotto = 1, Parte = 2, Impianto= 3);

- ValCTD: costi totali diretti. Sono quei costi direttamente attribuibili ai lavori che si realizzano;
- ValCTF: costi totali di fabbrica (es. costi capannone, costi manutenzione delle macchine);
- ValCTG: costi totali generali. Sono costi indipendenti dai lavori (es. gli stipendi dei dipendenti).

4.2 Estrazione dei dati

Il processo di estrazione dei dati è stato guidato da una serie di osservazioni, le quali hanno permesso che venissero estratti e considerati solo alcuni dei campi sopraelencati. Nella Premessa, è stato detto che i diversi clienti di Logica hanno una struttura specifica di produzione e, di conseguenza, una tipologia di prodotto finito differente. Il prodotto i cui preventivi sono stati analizzati è lo *shopper*. Le analisi sono state eseguite sul prodotto in cui l'azienda è maggiormente specializzata poiché, pur considerando il settore in cui l'azienda risulta essere più performante, il numero di preventivi non approvati è maggiore rispetto a quelli approvati (tra le N varianti del preventivo, solo una sarà approvata e, le altre N-1 non saranno approvate). Se si prendessero in considerazione settori in cui l'azienda è meno focalizzata si avrebbero classi maggiormente sbilanciate.

Dopo aver compreso il perché si sia scelto di analizzare quel particolare tipo di prodotto, è importante introdurre l'osservazione denominata *"Problema della granularità delle varianti"*.

Le varianti dei preventivi, come detto in precedenza, possono variare in relazione a: (i) tiratura (numero di copie del prodotto finito: es. cliente vuole sapere di quanto varia il costo in relazione al numero di copie desiderato); (ii) caratteristiche tecniche a livello granulare. Durante l'analisi, dunque, si potrebbe incorrere nel problema che le diverse varianti del medesimo preventivo differiscano fra loro per parametri granulari che, se non considerati,

potrebbero far in modo che lo stesso preventivo sia contemporaneamente approvato e non approvato. Allo stesso tempo, si ha l'esigenza di produrre un sistema che in futuro si possa adattare alle diverse tipologie di prodotto. Di conseguenza, non si possono considerare caratteristiche troppo specifiche. Dunque, in cooperazione con gli esperti del dominio, sono stati identificati i principali parametri granulari che distinguono le varianti del prodotto finito, rimanendo allo stesso tempo in un stato di genericità, così da non incorrere nel problema sopraesposto. Da tale analisi si evince che i parametri in questo contesto siano: la carta, lo spessore della carta, la coprenza e le colorazioni. Inoltre, si è utilizzata un'*euristica* tale per cui le varianti differiscano maggiormente in relazione alla tiratura.

I dati sono stati estratti sfruttando il linguaggio *SQL*. Di seguito si riportano le *query* utilizzate per l'estrazione dei dati.

```
01 | --Preventivi approvati
02 | SELECT H.JOBID, H.CodVar, H.CodReg, H.NumReg, H.Approved, TP.Nome AS
         NomeTipoProdotto, P.Copie, P.FmPag, P.LivDif, TPAR.Nome AS
        Parte, C.Nome AS Carta, PAR.GrCar, PAR.ColBi, PAR.ColVo, PAR.
        PcCopBi, PAR.PcCopVo, (select sum(Valctd+Valctf+Valctg) from
        jobrow where jobid=H.jobid and enabled=1) as CsTot
03 | FROM JOBHDR AS H INNER JOIN JOBPROD AS P ON H.JobID = P.JobID
04 | AND H.CodReg = P.CodReg
05 | AND H.NumReg = P.NumReg
06 | AND H.CodVar = P.CodVar
07 | INNER JOIN TPPROD AS TP ON P.Codice = TP.Codice
08 | INNER JOIN JOBPART PAR ON PAR.JobId=H.JobId AND PAR.idx=(SELECT min(
        idx) FROM jobpart WHERE jobid=H.jobid)
09 | INNER JOIN TIPIPAR TPAR ON PAR.codice=TPAR.codice
10 | INNER JOIN CARTA C ON C.Codice=PAR.CodCar
11 | WHERE H.CodReg LIKE 'PV%'
12 | AND H.CMNUMREG <> ''
13 | AND H.NumOrd <> ''
14 | AND H.codcli <> '00001'
15 | AND H.codcli <> ''
16 | AND P.Codice LIKE '300%'
17 | AND H.Approved = 1
18 | AND (SELECT COUNT(*) FROM JOBPROD WHERE JobID = H.JOBID) = 1
19 | AND (SELECT COUNT(*) FROM JOBPART WHERE JobID = H.JOBID) <= 6
20 | ORDER BY H.JOBID
```

```
01 | --Preventivi non approvati
02 | SELECT H.JOBID, H.CodVar, H.CodReg, H.NumReg, H.Approved, TP.Nome AS
       NomeTipoProdotto, P.Copie, P.FmPag, P.LivDif, TPAR.Nome AS
      Parte, C.Nome AS Carta, PAR.GrCar, PAR.ColBi, PAR.ColVo, PAR.
      PcCopBi, PAR.PcCopVo, (select sum(Valctd+Valctf+Valctg) from
      jobrow where jobid=H.jobid and enabled=1) as CsTot
03 | FROM JOBHDR AS H INNER JOIN JOBPROD AS P ON H.JobID = P.JobID
04 | AND H.CodReg = P.CodReg
05 | AND H.NumReg = P.NumReg
06 | AND H.CodVar = P.CodVar
07 | INNER JOIN TPPROD AS TP ON P.Codice = TP.Codice
08 | INNER JOIN JOBPART PAR ON PAR.JobId=H.JobId AND PAR.idx=(SELECT min(
      idx) FROM jobpart WHERE jobid=H.jobid)
09 | INNER JOIN TIPIPAR TPAR ON PAR.codice=TPAR.codice
10 | INNER JOIN CARTA C ON C.Codice=PAR.CodCar
11 | WHERE H.CodReg LIKE 'PV%'
12 | AND H.CMNUMREG = ''
13 | AND H.NumOrd = ''
14 | AND H.codcli <> '00001'
15 | AND H.codcli <> ''
16 | AND P.Codice LIKE '300%'
17 | AND H.Approved = 0
18 | AND (SELECT COUNT(*) FROM JOBPROD WHERE JobID = H.JOBID) = 1
19 | AND (SELECT COUNT(*) FROM JOBPART WHERE JobID = H.JOBID) <= 6
20 | ORDER BY H.JOBID
```

I campi presi in considerazione sono: JOBID, CodVar, CodReg, NumReg, Approved, Nome (TPPROD), Copie, FmPag, LivDif, Nome (TIPIPAR), Nome (CARTA), GrCar, ColBi, ColVo, PcCopBi, PcCopVo, ValCTD, ValCTF, ValCTG.

	JOBID	CodVar	CodReg	NumReg	Approved	NomeTipoProdotto	Copie	FmPag	LivDif	Parte	Carta	GrCar	ColBi	ColVo	PcCopBi	PcCopVo	CsTot
0	CLI1 BO01000241226	6	PV14	100433	1	Shopper in singolo	82500	36x56,5	1	Shopper	Symbol Bags	165	2	0	0	0	17856.066729
1	CLI1 BO01000241231	6	PV14	100434	1	Shopper in singolo	55000	44,5x72,5	1	Shopper	Symbol Bags	165	2	0	0	0	14920.985276
2	CLI1 BO01000241243	6	PV14	100437	1	Shopper in singolo	2200	70x73	1	Shopper	Symbol Bags	165	2	0	0	0	1569.150172
3	CLI1 BO01000241294	1	PV14	100449	1	Shopper in singolo	1050	52x82	1	Shopper	Forbags	170	5	0	100	0	473.289756
4	CLI1 BO01000241365	6	PV14	100462	1	Shopper in singolo	11000	44,5x98,5	1	Shopper	Symbol Bags	190	4	0	0	0	4837.771025
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
63615	CLI1 BO01000507594	23	PV20	706878	0	Shopper in singolo	300000	27,5x66,5	1	Shopper	Forbags	170	2	0	0	0	49036.778440
63616	CLI1 BO01000507599	9	PV20	706883	0	Shopper in singolo	10000	47x61,5	1	Shopper	Symbol Bags	165	4	0	0	0	4628.453716
63617	CLI1 BO01000507604	10	PV20	706883	0	Shopper in singolo	20000	47x61,5	1	Shopper	Symbol Bags	165	4	0	0	0	9016.775968
63618	CLI1 BO01000507605	23	PV20	706884	0	Shopper in singolo	1100	35,5x86,5	1	Shopper	Symbol Bags	190	1	0	0	0	466.717695
63619	CLI1 BO01000507606	23	PV20	706885	0	Shopper in singolo	1100	35,5x86,5	1	Fondi	Bianco/Grigio	490	0	0	0	0	187.463836

63620 rows × 17 columns

Figura 4.2: Dataset iniziale risultante dalla prima estrazione

4.3 Osservazioni sui dati

Fonti dei dati: le fonti dei dati sono interne all'azienda; sono cioè fonti operazionali.

Struttura dei dati: dati strutturati contenuti in un database relazionale.

Aspetto dei dati / Livello dei dati: colonne estratte che compongono le features del dataset.

Essendo JOB ID, Cod Var, Cod Reg e Num Reg dei campi ID, sono stati rimossi dal dataset, in quanto non utili nelle analisi predittive.

Approved: dato qualitativo (variabile categorica), livello nominale, variabile di output (obiettivo), variabile dicotomica.

NomeTipoProdotto: dato qualitativo (variabile categorica), livello nominale, variabile di input (predictor), variabile nominale.

Copie: dato quantitativo (variabile continua), livello dei rapporti, variabile di input (predictor), variabile numerica.

FmPag: dato qualitativo (variabile categorica), livello nominale, variabile di input (predictor), variabile nominale.

LivDif: dato qualitativo (variabile categorica), livello ordinale, variabile di input (predictor), variabile ordinale.

Parte: dato qualitativo (variabile categorica), livello nominale, variabile di input (predictor), variabile nominale.

Carta: dato qualitativo (variabile categorica), livello nominale, variabile di input (predictor), variabile nominale.

GrCar: dato quantitativo (variabile continua), livello dei rapporti, variabile di input (predictor), variabile numerica.

ColBi: dato quantitativo (variabile continua), livello dei rapporti, variabile di input (predictor), variabile numerica.

ColVo: dato quantitativo (variabile continua), livello dei rapporti, variabile di input (predictor), variabile numerica.

PcCopBi: dato quantitativo (variabile continua), livello dei rapporti, variabile di input (predictor), variabile numerica.

PcCopVo: dato quantitativo (variabile continua), livello dei rapporti, variabile di input (predictor), variabile numerica.

4.4 Preprocessing e Data cleansing

Il processo di pulizia dei dati è avvenuto, in parte, in parallelo alla fase di estrazione dei dati, in quanto è stato realizzato per mezzo di strumenti derivanti dallo Structured Query Language e, in parte, dopo l'estrazione tramite librerie del linguaggio di programmazione Python, come ad esempio Pandas.

Durante la realizzazione delle query – usate per estrapolare i dati di interesse dalla base di dati – sono state fatte delle osservazioni che hanno portato a rimuovere dei dati che avrebbero potuto alterare la correttezza dell'analisi:

- Per affermare che un preventivo è stato effettivamente approvato, sono stati presi in considerazione solamente quelli i cui campi (della tabella JOBHDR) CMNUMREG (numero di commessa) e NumOrd (numero ordine) sono non vuoti. Questo è dovuto al fatto che, se questi due campi sono occupati allora il preventivo è approvato. Per vari motivi, ad esempio, l'azienda conosce chi richiede l'ordine e, manda il prodotto in produzione senza far prima approvare il preventivo, potrebbe accadere che ci sono preventivi che sono presi e andati in produzione (hanno il numero di commessa) e vengono considerati come non approvati perché non hanno il numero ordine. Nel caso di preventivi non approvati sono stati considerati quelli i cui campi (della tabella JOBHDR) CMNUMREG (numero di commessa) e NumOrd (numero ordine) sono vuoti;

- Sono stati considerati solamente i preventivi il cui codice cliente è diverso da quello dell'azienda stessa. Questa considerazione è fondamentale perché l'azienda produce dei prodotti necessari per l'uso interno che non venderà ad altri clienti del settore. Se si considerassero pure questi preventivi, si accrescerebbe il numero di preventivi approvati (i quali non possono mai essere non approvati) il che, altererebbe il risultato finale dell'analisi;

- Sono stati considerati solamente i preventivi il cui codice cliente è non vuoto. Questa considerazione si è resa necessaria per eliminare i "preventivi di prova" realizzati dall'azienda stessa o, quei preventivi elaborati per la creazione di tariffe di vendita ma non elaborati su richiesta specifica di un cliente;
- Per precauzione sono stati rimossi i prodotti che contenevano più di sei parti, poiché il prodotto preso in considerazione per costruzione non può avere più di sei parti/componenti;
- Sono stati analizzati solamente i preventivi mono-prodotto, cioè sono stati rimossi dall'analisi quei prodotti composti da più elementi.

4.5 Gestione dei valori mancanti

Dopo aver estratto i dati, si è verificata la presenza di eventuali celle contenenti dei valori vuoti o, valori con un numero arbitrario di spazi o, ancora, valori *NULL*. In totale, all'interno dell'intero *dataframe*, sono stati identificati dieci valori *NaN*. Con il codice sotto riportato è stato possibile individuare le coordinate – in termini di righe e colonne – delle celle contenenti i valori mancanti.

```
01 | import pandas as pd
02 | import numpy as np
03 |
04 | df = pd.read_excel('preventivi.xlsx')
05 |
06 | df.replace(r'^\s*$', np.nan, regex=True, inplace=True)
07 |
08 | for i in df.columns:
09 |     tmp = df.loc[pd.isna(df[i]), :].index
10 |         if tmp.size != 0:
11 |             print("Colonna: ", i, '\n')
12 |             print("Righe: ", tmp.values,'\n')
13 |
14 | df.shape
15 |
```

```
[Out]:

Colonna: FmPag

Righe: [11138 11269 17962 18481 18482 18483 18484 18729 18730 18731]

(63620, 17)
-- fine out --

df.dropna(axis=0, how='any', inplace=True)

df.shape

[Out]:

(63610, 17)
```

Le diverse tipologie di valori mancanti prevedono differenti tipi di gestione. Nel caso in analisi, si tratta di valori *Missing Completely at Random* (MCAR), in quanto i dati sono mancanti indipendentemente dalle altre variabili e dai valori assunti da *FmPag* stessa. Poiché le righe contenenti i valori mancanti non costituiscono una percentuale significativa del dataset, si è deciso di eliminare le istanze incomplete.

4.6 Trasformazioni sui dati

Durante il processo di preparazione dei dati – finalizzato alla creazione del dataset su cui allenare e testare gli algoritmi – si possono eseguire delle trasformazioni sulle variabili come, ad esempio, operazioni di *scaling*, con l'obiettivo di rendere i dati adatti a certi algoritmi, i quali traggono vantaggio da tali manipolazioni. Esempi possono essere: la normalizzazione o la standardizzazione di variabili numeriche.

Il caso studio di questa tesi è stato affrontato sia con tecniche di apprendimento supervisionato, che non supervisionato. Nel primo caso, sono stati forniti agli algoritmi di *Machine Learning* sia i dati scalati che non scalati e, si è potuto constatare che la capacità predittive non variano; nel secondo

caso, si è osservato che l'algoritmo di clustering, essendo basato sul calcolo delle distanze, migliora le sue performance quando riceve come input dei dati standardizzati (vedi Figura 6.17).

4.7 Esplorazione delle variabili

Il processo di analisi dei dati comprende una fase di ispezione delle variabili. Tale fase è molto importante principalmente per tre ragioni: (i) si comprende la natura dei dati; (ii) si intuiscono in maniera preliminare le operazioni da applicare sui dati; (iii) si ottengono informazioni propedeutiche alla comprensione del significato dei risultati prodotti dagli algoritmi. Di seguito, si descrivono le tecniche adottate a tal fine.

4.7.1 Analisi univariata

L'analisi univariata consente di esplorare le variabili considerandole una per volta.

Nel caso di variabili continue, si sfruttano delle tecniche statistiche che permettono di ottenere delle informazioni dai dati. A livello visuale, tali informazioni, si rappresentano tramite i *box-plot* altrimenti noti come diagrammi a scatola e baffi. La Figura 4.3 mostra i risultati dell'analisi univariata eseguita sulle variabili numeriche del dataset.

Relativamente alle variabili categoriche, si possono adottare tecniche di conteggio del numero di esempi di ogni categoria, o anche percentuali sul totale degli esempi. Le metodologie di rappresentazione utilizzate sono gli istogrammi e i grafici a barre. La Figura 4.4 espone i risultati derivanti dall'analisi univariata applicata alla variabile categorica *Label*. L'informazione restituita da tale variabile è che le classi sono sbilanciate. Il numero di esempi di preventivi approvati (15 328) è inferiore a quelli non approvati (48 282).

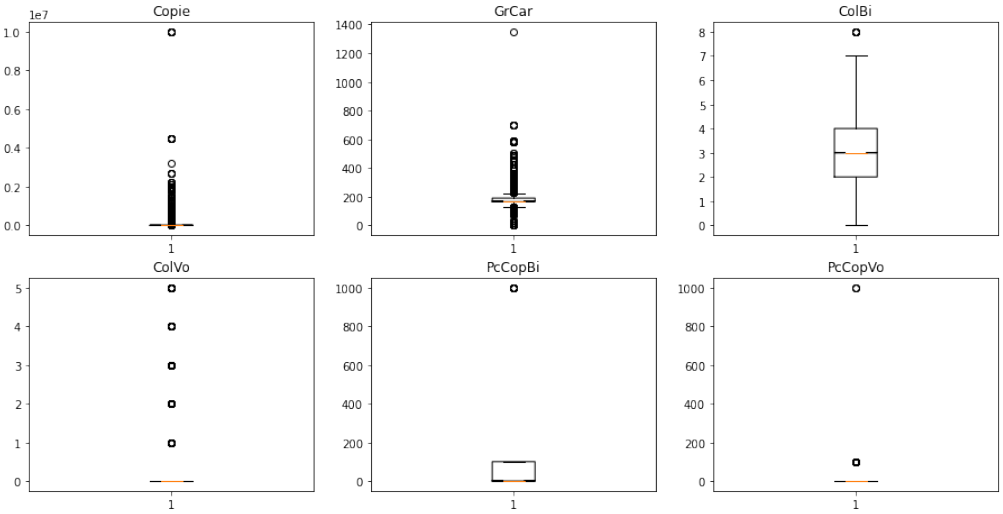


Figura 4.3: *Box-plot* delle variabili numeriche del dataset

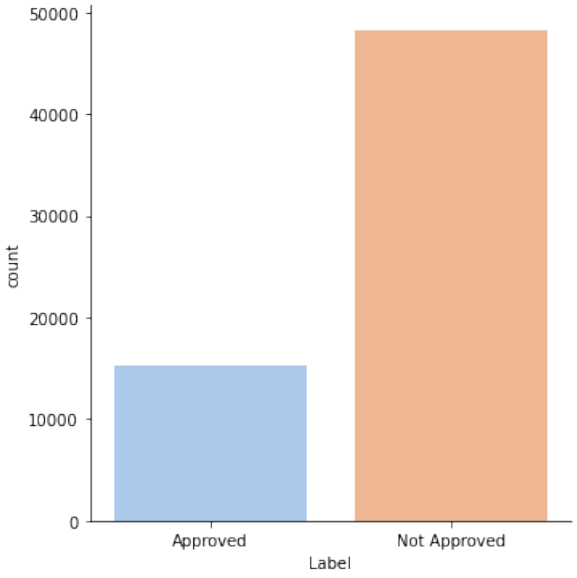


Figura 4.4: Grafico a barre della variabile *Label*

4.7.2 Analisi bivariata

Durante la fase di esplorazione delle variabili, sono state adottate anche tecniche di analisi bivariata. In questo caso, si analizzano due variabili per volta e, i possibili confronti che si realizzano sono: (i) due variabili continue; (ii)

due variabili categoriche; (iii) una variabile continua e una variabile categorica. Nel caso studio di questo elaborato, tale analisi è avvenuta dopo la fase di riduzione della dimensionalità del dataset (discussa nella sezione 4.11 di questo capitolo), in quanto non si è in grado di rappresentare dati n-dimensionali. A valle del processo di *features selection* sono state identificate come variabili rilevanti: *Copie*, *LivDif*, *Carta speciale (Specificare)* e, su queste, si è condotta l'analisi bivariata. La prima di queste è una variabile continua, le altre due sono categoriche.

La Figura 4.5 mostra la distribuzione del numero di copie in relazione al livello di difficoltà di realizzazione del prodotto.

La Figura 4.6 mette in luce la distribuzione del numero di copie relativamente all'uso di una carta speciale.

La Figura 4.7 mostra la distribuzione del numero di copie in relazione all'approvazione o meno dei preventivi e, al livello di difficoltà di realizzazione.

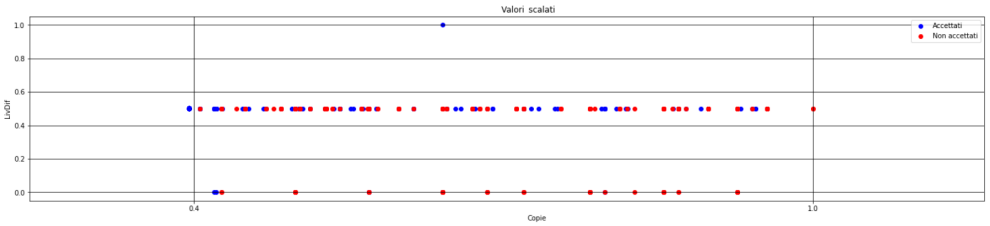


Figura 4.5: Grafico a dispersione delle variabili *Copie* e *LivDif*

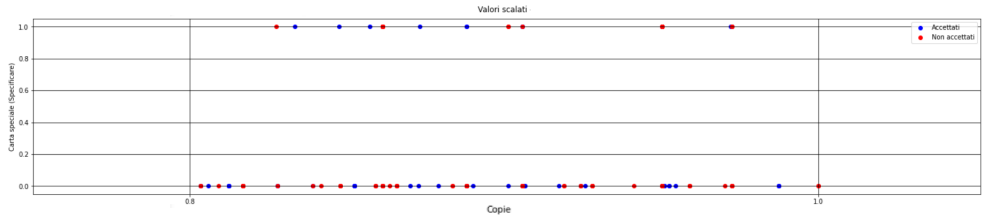


Figura 4.6: Grafico a dispersione delle variabili *Copie* e *Carta speciale (Specificare)*

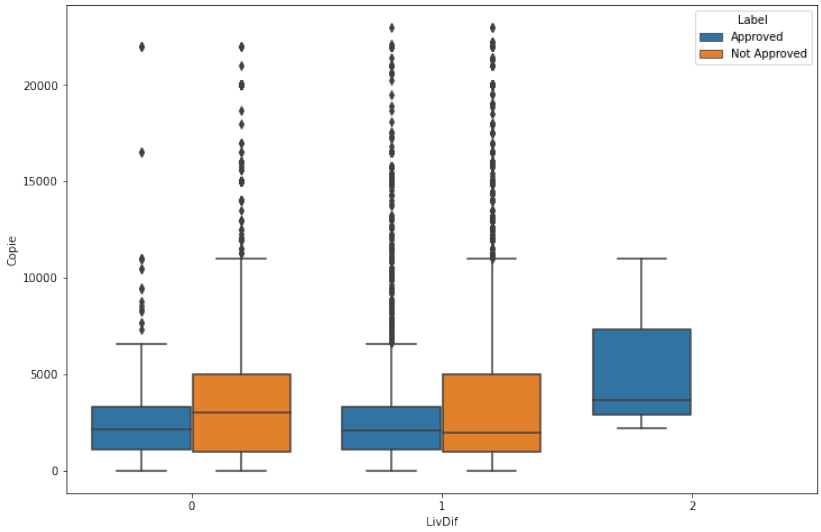


Figura 4.7: *Box-plot* della variabile *Copie*

Dai primi due spezzoni di grafici, si evince che i preventivi sono suddivisi in intervalli a formare dei *clusters*. In questa fase, si è potuto meglio comprendere il perché: algoritmi come il *Decision tree classifier* e *K-nearest neighbors* riescono a percepire il *pattern* tra i dati a differenza di altri, come ad esempio, il *Support vector machine*. Il terzo grafico è utile alla fase di studio dei valori anomali (4.8).

4.7.3 Analisi della correlazione lineare

Ulteriori considerazioni sono state tratte a seguito dell'analisi della correlazione lineare tra le variabili del dataset. I coefficienti di correlazione sono una misura dell'intensità con cui due variabili si *muovono* insieme. Un valore tendente ad 1 indica correlazione positiva (al crescere di una variabile l'altra cresce), un valore tendente a -1 indica correlazione negativa (al crescere di una variabile l'altra decresce), un valore tendente a 0 indica assenza di correlazione. Di seguito si riportano i risultati di tale analisi.

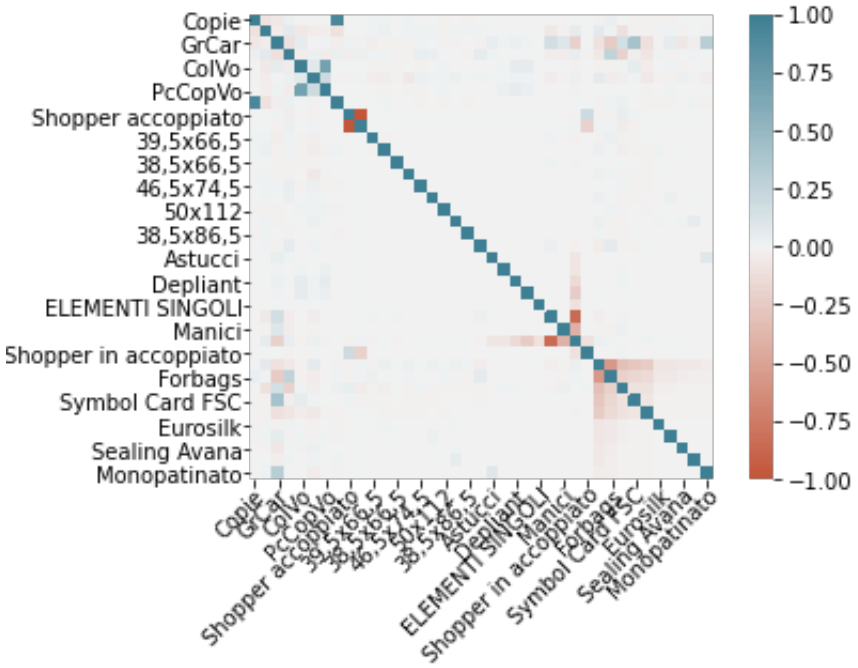


Figura 4.8: *Correlation heatmap*

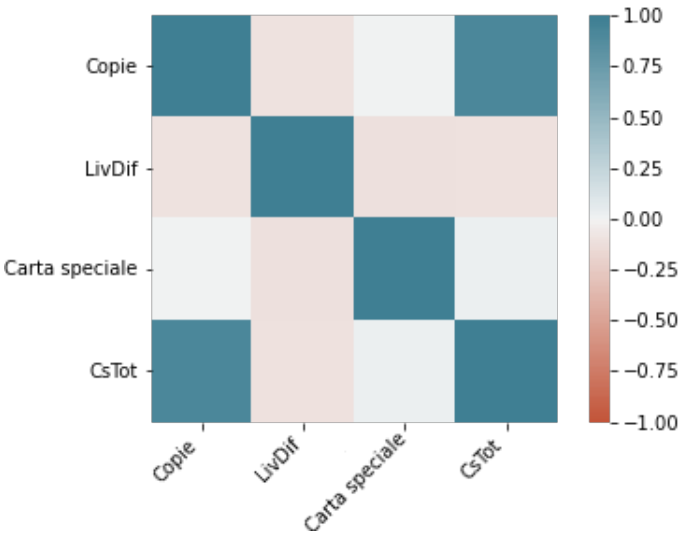


Figura 4.9: *Correlation heatmap*

Dalla prima rappresentazione grafica si desume che la maggior parte delle variabili sono incorrelate fra loro. La seconda, realizzata a seguito del processo di riduzione della dimensionalità del dataset, rimarca la correlazione

lineare positiva che sussiste fra le variabili *Copie* e *CsTot* (costi totali). In un primo momento si è proceduto con una verifica sperimentale, ma la relazione: "al crescere del numero di copie aumentano i costi di produzione" è abbastanza prevedibile. Le variabili linearmente dipendenti devono essere eliminate dal dataset, in quanto gli algoritmi di ML ne soffrono. Di conseguenza, la variabile *CsTot* è stata rimossa. Ancora una volta, mediante una verifica sperimentale, è stato dimostrato come le capacità predittive del *Decision tree classiffier* peggiorino a causa della presenza di variabili linearmente dipendenti (capitolo 6 Figura 6.10).

4.8 Gestione dei valori anomali

Nel corso del processo di analisi univariata e bivariata sono stati prodotti diversi risultati. Il grafico riportato nella Figura 4.3 mette in evidenza il fatto che alcune features (es. Copie, GrCar, ColVo) sono caratterizzate dalla presenza di molti outliers.

Tuttavia, effettuando una pulizia dei dati tramite lo scarto interquartile[2] e, addestrando il classificatore migliore (*Decision tree classifier*) su tale dataset (sia ridotto che con tutte le features), si è potuto constatare che le prestazioni non sono migliorate. Questo è dovuto al fatto che le features non contengono valori anomali, bensì è la natura stessa dei dati che li rende così distribuiti. Di conseguenza, è errato chiamarli outliers.

La Figura 4.7 rappresenta la distribuzione del numero di copie a seguito della pulizia dei dati tramite lo scarto interquartile, ma come si nota le istanze sono ugualmente sparse essendo per natura così distribuite.

[2]Lo scarto interquartile (IQR) è la differenza tra il terzo ($Q3$) e il primo ($Q1$) quartile. Sono state considerate le sole istanze del dataset il cui parametro *Copie* fosse maggiore del limite inferiore $Q1 - 1.5 \cdot IQR$ e, inferiore rispetto al limite superiore $Q3 + 1.5 \cdot IQR$.

4.9 Codifica di variabili categoriche

Molti algoritmi di apprendimento automatico non possono operare direttamente sui dati categorici; richiedono che tutte le variabili di input e di output siano numeriche. Ciò significa che i dati categoriali devono essere convertiti in una forma numerica [22].

Il processo di *one-hot encoding* prevede che si creino tante colonne quante sono le categorie e, per ogni riga del dataset, si pone a 1 il valore della colonna corrispondente alla categoria, 0 altrimenti. La Figura 4.10 mostra per via grafica il processo appena descritto.

Le variabili del dataset interessate da questo processo sono: NomeTipoProdotto e Parte.

Nel dataset sono presenti ulteriori variabili categoriche caratterizzate dalla presenza di un elevato numero di categorie al loro interno. Di conseguenza, per evitare di avere un dataset con un elevato numero di features, è stata utilizzata una tecnica simile al *one-hot encoding*, che considera solamente le dieci categorie più ricorrenti. Questo metodo è stato applicato da alcuni ricercatori di *IBM Research* durante la competizione *KDD Cup* [23].

Le variabili trattate con questa tecnica sono: FmPag e Carta.

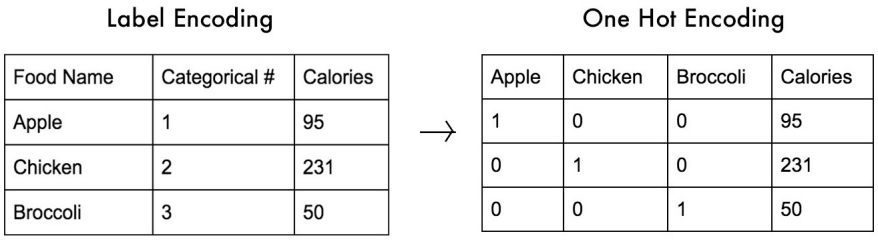


Figura 4.10: Processo *one-hot encoding*

4.10 Classi sbilanciate

A seguito del processo di analisi univariata della variabile categorica *Label*, è stato possibile rappresentare graficamente il totale degli esempi delle due

classi di preventivi (Figura 4.4). Da tale grafico, si evince che si è in presenza di un dataset sbilanciato. Il numero di esempi della classe "preventivo non approvato" è maggiore di quella "preventivo approvato". Questo è dovuto al fatto che, dello stesso preventivo, possono essere realizzate più varianti; delle quali solamente una verrà approvata (vedi problema granularità varianti descritto nella sezione 4.2 di questo capitolo).

Lo sbilanciamento delle classi potrebbe avere delle forti ripercussioni sul modello. Il risultato delle predizioni potrebbe essere sempre la classe maggioritaria. Di conseguenza, si è intervenuti in fase di preparazione dei dati con tecniche di *oversampling* e *undersampling* (applicate esclusivamente al training-set).

La strategia utilizzata è stata la seguente:

i. A livello di modellazione, sono stati testati diversi algoritmi addestrati su parte del dataset non bilanciato, per capire se almeno uno di essi fosse stato in grado di cogliere il pattern sottostante alla classe minoritaria [24];

ii. Sono stati testati diversi algoritmi addestrati sul training-set bilanciato con tecniche di *undersampling* con strategia di classe maggioritaria;

iii. Sono stati testati diversi algoritmi addestrati sul training set bilanciato con tecniche di *oversampling* con strategia di classe minoritaria.

A seguito del bilanciamento del dataset, gli esiti prodotti dal miglior algoritmo (*Decision Tree Classifier*) sono rimasti identici (vedi Figure 6.6 e 6.9). Non avendo ottenuto significativi miglioramenti dal bilanciamento del set di dati, si è deciso di utilizzare quello non bilanciato per evitare di introdurre rumore o causare problemi di *underfitting* o di *overfitting*.

4.11 Riduzione della dimensionalità

Quando si parla di processo di riduzione della dimensionalità del dataset, si intende l'attività di riduzione del numero di variabili da fornire agli algoritmi

di ML, senza diminuirne le capacità predittive oppure, l'attività di riduzione del numero di variabili minimizzando la perdita di informazioni al fine di poter rappresentare i dati in uno spazio con meno dimensioni.

Questo processo talvolta si rende necessario, in quanto le prestazioni degli algoritmi possono calare al crescere del numero di features. Un'ulteriore motivazione – in accordo con il rasoio di Occam[3] – è data dal fatto che i modelli che producono gli algoritmi devono essere semplici da comunicare; la crescita del numero di *features* implica complessità [25].

Nella sezione risultati si può appurare che nel particolare caso studio di questo elaborato, si ottengono effettivi miglioramenti a valle della riduzione della dimensionalità (Figura 6.9). L'attività descritta in questa sezione è stata condotta combinando tecniche, risultati ottenuti e algoritmi di apprendimento automatico utili a tale scopo.

4.11.1 Features selection

Un primo approccio di risoluzione del problema descritto sopra, prevede l'uso di diverse tecniche non complementari fra loro [25]. Pertanto, il risultato esposto in questa sotto-sezione, deriva dalla combinazione delle seguenti metodologie:

i. A seguito dell'analisi dei diversi classificatori, si è constatato che il *Decision tree classifier* ha prodotto i risultati migliori, conseguentemente, è stato deciso di determinare il numero ottimo di *features* – ovvero quello per cui si ha l'accuracy massima – in relazione a tale classificatore. Dopodiché, è stato utilizzato l'algoritmo di selezione *Recursive Feature Elimination* (RFE) combinato ad una tecnica di *cross-validation* per

[3] Il rasoio di Occam [...] è un principio metodologico che, tra più ipotesi per la risoluzione di un problema, indica di scegliere, a parità di risultati, quella più semplice [...]. Rasoio di Occam. (2021, April 26). In Wikipedia. `https://it.wikipedia.org/wiki/Rasoio_di_Occam`

l'addestramento e successivo test. La Figura 4.11 mostra i risultati di questa prima fase;

ii. Si è proseguito visualizzando i valori di importanza che sono stati assegnati alle features dal classificatore durante il suo processo di addestramento, al fine di poter determinare le due ottime;

iii. Ancora, è stato sfruttato un ulteriore algoritmo di Machine Learning: *LassoCV*. Il risultato dell'importanza delle features è stato rappresentato tramite il grafico a barre presente nella Figura 4.12;

iv. Infine, si è fatto ricorso all'algoritmo per la selezione della features: *SelectKBest con score_func di default (f_classif)*. L'esecuzione di tale algoritmo ha prodotto i seguenti risultati: ['Copie' 'LivDif' 'GrCar' 'Symbol Bags' 'Forbags' 'Carta speciale (Specificare)' 'Kraft Avana'] – Accuracy con 7 features: 0.94;

v. Lo stesso algoritmo è stato avviato in seguito alla pulizia del dataset tramite lo scarto interquartile (IQR). I risultati ottenuti sono i seguenti: ['Copie' 'LivDif' 'Carta speciale (Specificare)'] – Accuracy con 3 features: 0.94.

Dopo una serie di considerazioni finali, combinando gli esiti ottenuti dai vari algoritmi, si è deciso di utilizzare le features 'Copie', 'LivDif' e 'Carta speciale (Specificare)'. Dai risultati presenti nelle Figure 6.6 e 6.9 del capitolo 6, si può constatare che, riducendo le dimensionalità del dataset, si ottiene un miglioramento delle prestazioni del classificatore; il tasso dei *true-positive* cresce.

È importante notare che, le prestazioni degli algoritmi con le seguenti combinazioni di features: 'Copie' – 'LivDif' e 'Copie' – 'Carta speciale (Specificare)' sono le stesse rispetto alla combinazione di tutte e tre le features. Tuttavia, si è scelto di usare sia 'LivDif' che 'Carta speciale (Specificare)' insieme alla feature 'Copie' (e non solo una delle due), in quanto gli algo-

ritmi di ML non sono infallibili e, si è voluto garantire un miglior livello di separazione tra i clusters.

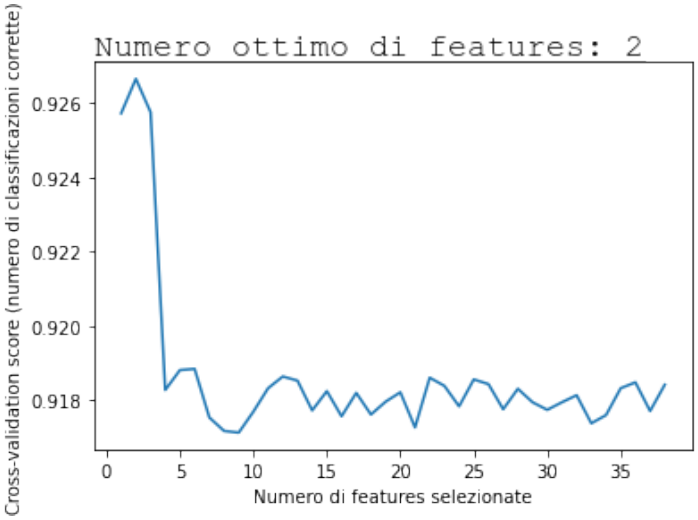


Figura 4.11: Risultato dell'algoritmo *Recursive Feature Elimination*

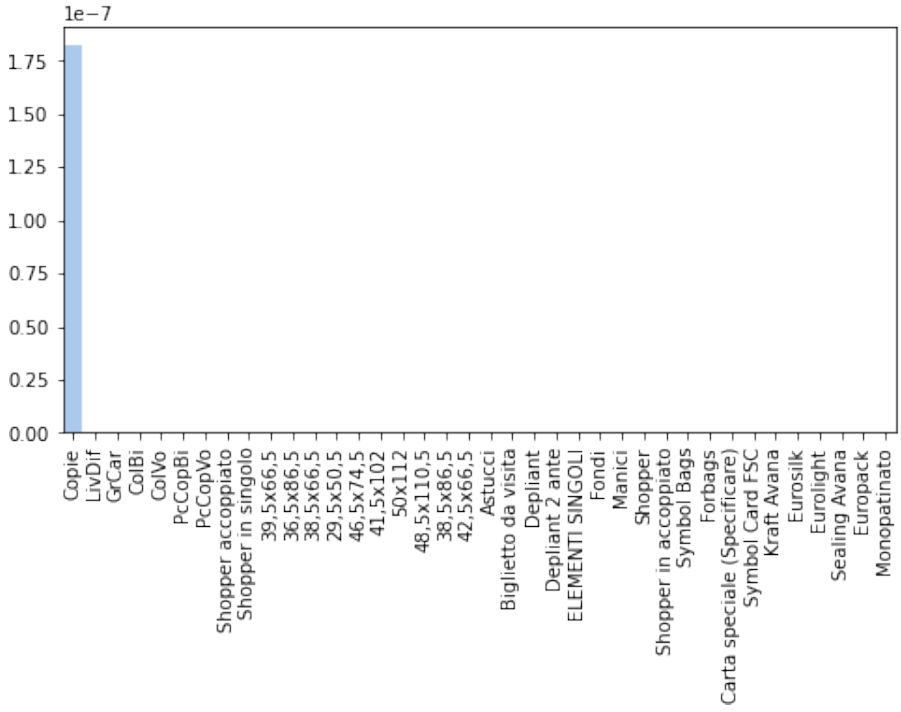


Figura 4.12: Risultato dell'algoritmo *LassoCV*

4.11.2 Principal Component Analysis

La *Principal Component Analysis* (PCA) è una tecnica di riduzione della dimensionalità di dataset n-dimensionali. L'obiettivo è quello di garantire una migliore comprensione dei dati preservando l'informazione contenuta nelle variabili originali.

Le principali caratteristiche dell'algoritmo sono: (i) è una tecnica di riduzione della dimensionalità *lineare*; (ii) è un algoritmo deterministico; (iii) preserva la struttura globale dei dati; (iv) i valori anomali hanno influenza durante la sua esecuzione [26, 27].

I risultati, derivanti dall'esecuzione di tale algoritmo sul dataset del caso studio di questa tesi, sono riportati nella sottosezione 6.1.1 del capitolo 6.

Di seguito, si riportano i passi previsti durante l'esecuzione dell'algoritmo. Nell'Appendice si riportano una serie di informazioni che aiutano a comprendere il perché l'algoritmo esegua certe operazioni.

L'obiettivo è quello di passare da uno spazio n-dimensionale ad uno k-dimensionale con $k < p$. Per comodità, il dataset viene espresso in forma matriciale [28]:

1. Al passo uno, l'algoritmo intraprende una di queste due possibili azioni: (i) standardizza i dati contenuti nella matrice X (operazione necessaria che riporta i dati di input nella stessa scala, in quanto nelle colonne potrebbero essere presenti quantità con unità di misure differenti); (ii) calcola la media di ogni colonna e, sottrae le medie in ogni colonna elemento per elemento (i dati vengono traslati e centrati nell'origine). In altre parole, si porta la matrice nella forma *Mean Deviation* (ovvero a media nulla – operazione funzionale al calcolo della matrice di covarianza).

   Il risultato di una delle due operazioni si indica con $\tilde{X}$;

2. Nel secondo passo, calcola la matrice di covarianza Σ_X, ovvero, una matrice simmetrica. Ogni elemento sulla diagonale principale rappresenta il valore di varianza della variabile X_i, mentre, quelli fuori dalla

diagonale, rappresentano le covarianze tra le variabili X_i e X_j con $j \neq i$. I valori al di fuori della diagonale possono essere: positivi (al crescere di X_i, cresce X_j; negativi (al crescere di X_i, decresce X_j); zero (X_i e X_j sono statisticamente indipendenti) [29];

3. Il terzo passo dell'algoritmo – si veda la dimostrazione A.1 per la comprensione di questo passo – consiste nel calcolo degli autovalori $\lambda_1, \ldots, \lambda_n$ e relativi autovettori $v_1, \ldots, v_n$ di Σ_X.

   Gli autovettori sono il sistema di riferimento migliore per rappresentare i dati di partenza e, gli autovalori sono un indice di variabilità che rivela gli autovettori da considerare (le componenti che variano molto forniscono informazione)[4]. Si pongono gli autovalori in ordine decrescente e si considerano solo i primi k autovettori corrispondenti ai primi k autovalori;

4. Il quarto passo richiede di creare le componenti principali sfruttando la relazione lineare $Y = P\tilde{X}$.

   Y è la matrice che contiene le componenti principali; le matrici P e $\tilde{X}$ sono rispettivamente: la matrice ortonormale di autovettori selezionati al punto precedente e, la matrice scalata al punto uno.

[4] Avendo un dataset di CD con diverse caratteristiche: numero delle tracce, artista, genere, ... e, dovendo condurre un'analisi su quali siano i CD più venduti, non si considerano proprietà che forniscono informazioni ridondanti (correlate) come, ad esempio, la feature numero di tracce. Di fatto, quasi tutti gli esempi hanno 14/15 tracce e, quindi, una feature con bassa varianza fa apparire i CD tutti uguali. Diversamente, considerare la feature genere musicale, caratterizzata da un'alta varianza, potrebbe portare a un risultato più rilevante.

4.11.3 t-Distributed Stochastic Neighbor Embedding

T-Distributed Stochastic Neighbor Embedding (t-SNE) è un algoritmo di apprendimento automatico non supervisionato per la riduzione della dimensionalità di set di dati. Gli obiettivi di questo algoritmo sono: migliorare l'interpretabilità dei dati e minimizzare la perdita di informazioni durante il processo di riduzione.

Le sostanziali differenze con l'algoritmo PCA sono: (i) è una tecnica di riduzione della dimensionalità *non lineare*; (ii) modella gli oggetti nel nuovo spazio garantendo il mantenimento delle distanze dello spazio originale; (iii) preserva la struttura locale dei dati; (iv) i valori anomali non hanno influenza durante la sua esecuzione [26].

I risultati, derivanti dall'esecuzione di tale algoritmo sul dataset del caso studio di questa tesi, sono riportati nella sottosezione 6.1.2 del capitolo 6.

Di seguito viene riportata una trattazione ad alto livello dei passi dell'algoritmo. Le informazioni e le formule sono state attinte dalla pubblicazione ufficiale [30] e, da un articolo disponibile nella piattaforma *Medium* [31].

1. Il primo passo dell'algoritmo consiste nel calcolo della somiglianza per ogni coppia di punti nello spazio con N dimensioni. Dalle fonti citate sopra, si è ottenuto che "la somiglianza del punto x_j con il punto x_i è la probabilità condizionata $p_{j|i}$, che x_i scelga x_j come suo vicino."; tale probabilità è data da:

$$p_{j|i} = \frac{exp\left(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} exp\left(-\left\|x_i - x_k\right\|^2 / 2\sigma_i^2\right)} \tag{4.1}$$

   Il calcolo della somiglianza viene eseguito sfruttando una distribuzione normale centrata nel punto x_i (numeratore della 4.1), quindi a punti distanti da x_i vengono assegnati valori bassi di somiglianza e, a punti vicini a x_i vengono assegnati valori alti di somiglianza. Tuttavia, questi sono valori "non scalati". Prima di proseguire con il passo due dell'algoritmo, i risultati ottenuti fino a questo momento devono essere

scalati; in maniera tale che la loro somma valga esattamente 1. Questo passaggio si rende necessario perché, in genere, ci possono essere dei clusters più e meno densi. La densità del cluster ha effetto sulla deviazione standard σ_i (valore riassuntivo della dispersione dei singoli punti). Se il cluster è denso – i punti sono poco dispersi – allora il valore di σ_i è basso. Al contrario, se il cluster è meno denso – i punti sono molto dispersi – allora il valore di σ_i è alto. A sua volta, la deviazione standard ha effetto sull'ampiezza della curva di Gauss.

Dunque, nel caso di cluster poco densi, si hanno gaussiane "basse" e "larghe". I valori di probabilità assunti (asse y) sono più piccoli rispetto al caso di gaussiane di cluster più densi, caratterizzate dall'essere "alte" e "corte" (Figure 4.13 e 4.14).

L'operazione di normalizzazione garantisce la risoluzione di questo problema seppur in presenza di gaussiane con larghezze differenti. Questa operazione di scaling si ottiene dividendo il valore di somiglianza rispetto alla somma di tutte le somiglianze (denominatore della 4.1);

L'ultima considerazione necessaria al termine del passo uno è che: quando si calcolano i valori di $p_{j|i}$ e $p_{i|j}$, questi risultano essere differenti a causa del fatto che i punti provengono da distribuzioni differenti. Il valore di somiglianza considerato al termine del processo è dato da:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{4.2}$$

2. Nel secondo passo, l'algoritmo distribuisce in maniera casuale i punti all'interno del nuovo spazio con dimensioni ridotte e, procede con il calcolo delle somiglianze dei punti. In questo contesto non si sfrutta una gaussiana come nel caso precedente, bensì una distribuzione *t di Student* (la Figura 4.15 mostra le differenze con la distribuzione normale usata nel passo uno). Tale scelta, è motivata dal fatto che le code della gaussiana sono "corte". Infatti, utilizzando questa distribuzione, non si riuscirebbero a distinguere i clusters, in quanto si avrebbero

punti schiacciati e ammassati. Infine, i valori ottenuti vengono scalati come nel passo precedente. La formula che permette di calcolare le somiglianze dei punti nel nuovo spazio:

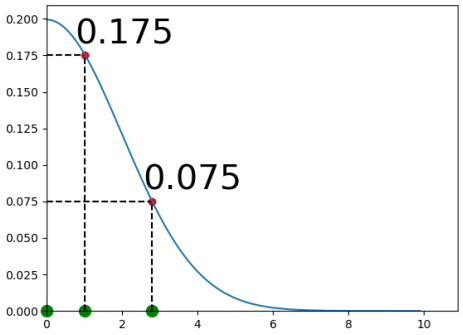
$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}} \tag{4.3}$$

3. Il terzo e ultimo passo dell'algoritmo consiste nello spostare i punti nel nuovo spazio ridotto fin quando i valori calcolati al punto due non sono simili a quelli calcolati nel punto uno. L'algoritmo minimizza la *divergenza di Kullback-Leibler*[5] delle due distribuzioni di somiglianze P e Q (formula 4.4) tramite il metodo di discesa del gradiente (formula 4.5 – tecnica che consente di determinare i punti di massimo e minimo di una funzione di più variabili) [32, 33, 34]. Il gradiente descrive la forza di repulsione/attrazione tra i punti e le relative direzioni.

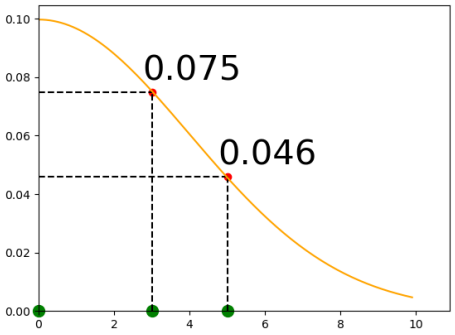
$$C = D_{\mathrm{KL}}(P\|Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right) \tag{4.4}$$

$$\frac{\delta C}{\delta y_i} = 4 \sum_j \left(p_{ij} - q_{ij}\right)\left(y_i - y_j\right)\left(1 + \|y_i - y_j\|^2\right)^{-1} \tag{4.5}$$

[5] In teoria della probabilità [...] la divergenza di Kullback–Leibler [...] è la misura dell'informazione persa quando Q è usata per approssimare P [...]. Tipicamente P rappresenta la "vera" distribuzione di dati [...]. La misura Q tipicamente rappresenta una teoria, modello, descrizione, o approssimazione di P. Divergenza di Kullback-Leibler. (2020, June 11). In Wikipedia. `https://it.wikipedia.org/wiki/Divergenza_di_Kullback-Leibler`



(a) Gaussiana centrata in x_i con σ_i piccolo



(b) Gaussiana centrata in x_i con σ_i grande

Figura 4.13: I valori di somiglianza (asse y) in (a) sono più alti rispetto a quelli in (b) a causa di σ_i

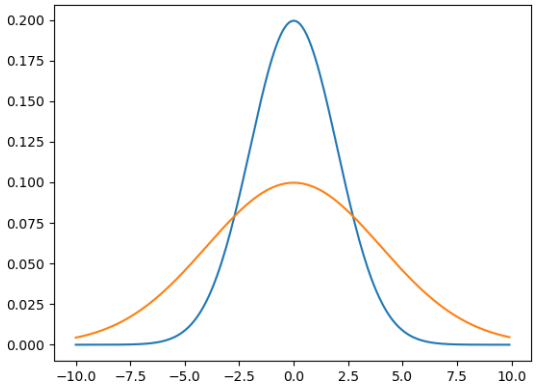


Figura 4.14: Gaussiane messe a confronto

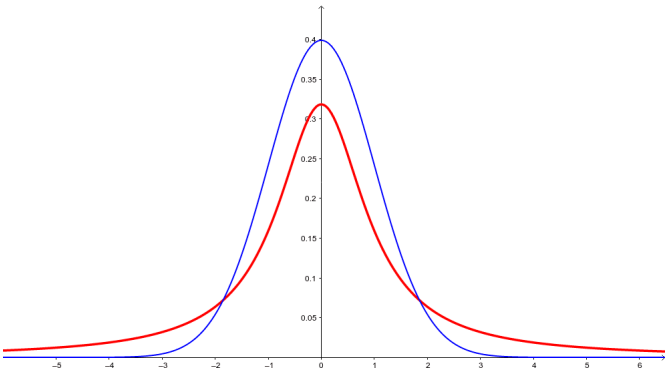


Figura 4.15: Distribuzione t-Student (rossa) vs. Distribuzione normale (blu)

Capitolo 5

Processi di addestramento

5.1 Algoritmi di apprendimento automatico supervisionato

In questa sezione si descrivono i processi di addestramento e il funzionamento degli algoritmi di apprendimento automatico supervisionato da un punto di vista teorico-procedurale-matematico.

5.1.1 Decision tree classifier

Uno degli algoritmi che ha prodotto dei buoni risultati è stato l'albero delle decisioni, in inglese *Decision Trees*. Esistono diversi algoritmi di alberi decisionali: *ID3*, *C4.5*, *C5.0*, e *CART*. La libreria di ML usata (Scikit-learn) implementa una versione ottimizzata dell'algoritmo *CART* (*Classification and Regression Trees*) [35].

In generale, se l'albero delle decisioni classifica le istanze in categorie, si parla di *Classification Tree*; altrimenti, se effettua delle predizioni sui numeri, si parla di *Regression Tree*.

Al termine dell'addestramento l'algoritmo produce un modello dei dati. L'albero risultante è composto da (vedi Figura 5.1):

i. Nodo radice;

ii. Archi: rappresentano le decisioni;

iii. Nodi interni: i cui archi sono sia entranti che uscenti. Rappresentano dei test sugli attributi;

iv. Nodi foglia: i cui archi sono solamente entranti e non uscenti. Tali nodi sono dotati dell'etichetta di una classe.

Il funzionamento dell'algoritmo *CART* è basato sull'uso di una metrica denominata *impurity*. Ad ogni nodo dell'albero si affianca un set di record del dataset che viene poi suddiviso a valle di un test su una feature. In aggiunta, al nodo dell'albero, si associa una quantità di purezza. L'impurity fornisce una misura dell'omogeneità dei campioni considerati. Se tutti i campioni sono omogenei, allora appartengono alla stessa classe [36].

Esistono diversi metodi che consentono di misurare la purezza dei campioni: (i) Entropia; (ii) Indice di Gini; (iii) Tasso d'errore (Misclassification).

"L'entropia è la quantità di informazioni necessarie per descrivere accuratamente il campione." [36]. Il tasso d'errore fornisce una misura di classificazione errata. Tale metrica si adopera in contesti multi-classe.

L'indice di Gini assume un valore $\in [0, 1]$; 0 quando i campioni sono omogenei, 1 nel caso opposto. Tramite questa metrica è possibile stabilire quale feature fornisce la suddivisione ottimale dei dati.

La costruzione dell'albero parte dalla valutazione dell'indice Gini di tutte le features. Dopo aver individuato la feature del nodo radice, con relativa condizione, si procede calcolando l'indice di Gini dei nodi foglia, derivanti dalla suddivisione del set di dati associato a tali nodi. Nell'Appendice è stata riportata una spiegazione più dettagliata del calcolo e delle formule (vedi A.2).

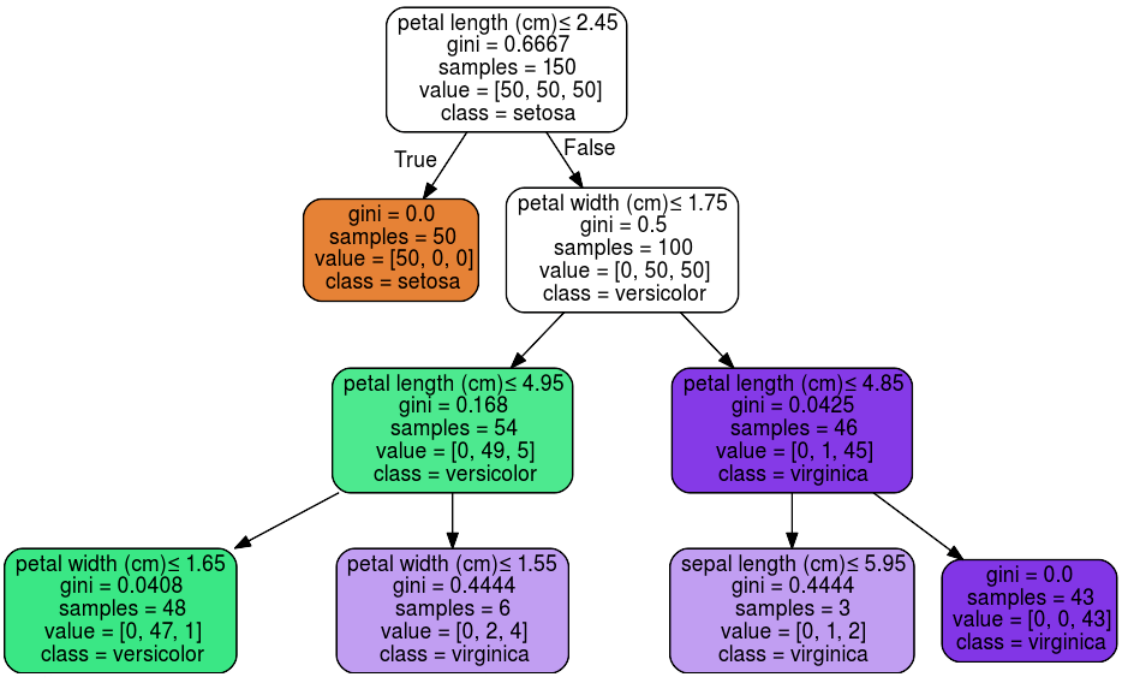


Figura 5.1: Esempio di albero delle decisioni creato sul dataset iris

5.1.2 K-nearest neighbors

Il *K-Nearest Neighbors* (KNN) è un algoritmo di apprendimento automatico utilizzato negli ambiti di *pattern recognition* e classificazione. Malgrado la sua semplicità, questo algoritmo, si è prestato molto bene al caso studio oggetto di questa tesi. Fornendo all'algoritmo un set di dati composto da n *training-vectors*, questo identifica la classe dell'oggetto da catalogare tramite il calcolo della distanza tra tutti i vettori del dataset – dotati di un'etichetta – e il nuovo vettore in ingresso. Dopodiché, l'algoritmo seleziona i k vettori più vicini e, tramite un sistema di voti con strategia di maggioranza, assegna una classe al nuovo elemento (Figura 5.2).

Le funzioni che permettono di calcolare la distanza sono: (i) Distanza Euclidea; (ii) Distanza Minkowski; (iii) Distanza di Hamming.

$\mathbb{R}^n$ è uno spazio vettoriale su $\mathbb{R}$. Gli elementi di tale spazio prendono il nome di vettori: $X = \{x_1, \ldots, x_n\}$.

La Distanza Euclidea tra due vettori $X = \{x_1, \ldots, x_n\}$ e $P = \{p_1, \ldots, p_n\}$ è definita da:

$$\begin{aligned} d(X,P) &= \sqrt{(x_1 - p_1)^2 + (x_2 - p_2)^2 + \cdots + (x_n - p_n)^2} \\ &= \sqrt{\sum_{k=1}^{n} (x_k - p_k)^2}. \end{aligned} \tag{5.1}$$

Un'ulteriore possibile metodologia di identificazione della classe di appartenenza del vettore di ingresso, oltre al sistema di voti, si basa sul calcolo di una *pseudo probabilità condizionata*:

$$P\left(y = j \mid X = x\right) = \frac{1}{K} \sum_{i \in \mathcal{A}} I\left(y^{(i)} = j\right) \tag{5.2}$$

k è un intero che indica il numero dei vicini, $\mathcal{A}$ è il set dei k vicini e $I\left(y^{(i)} = j\right)$ è una funzione che fornisce in output il valore 1, se la classe del vicino $x^{(i)}$ è $y^{(i)} = j$, 0 altrimenti. Infine, l'algoritmo seleziona la classe con il valore di probabilità più alto.

In realtà, gli output della 5.2 non sono valori di probabilità, bensì, porzioni di elementi che soddisfano una determinata condizione.

Il parametro k solitamente assume un valore maggiore di uno. Tuttavia, esiste un caso speciale in cui il parametro k vale esattamente uno. In tale circostanza, l'algoritmo crea una partizione di *Voronoi* dello spazio. Ovvero, ogni vettore del dataset definisce una regione che gode della seguente proprietà (*decision boundary*) [37]:

$$R_i = \{x : d\left(x, x_i\right) < d\left(x, x_j\right), i \neq j\} \tag{5.3}$$

La regione R_i contiene tutti i punti dello spazio più vicini a x_i che a qualsiasi altro punto x_j del training set. Ad ogni nuovo elemento che cade

all'interno della regione R_i, viene assegnata l'etichetta del vettore x_i che ha determinato quella partizione dello spazio.

Parte delle informazioni riportate in questa sezione, sono state apprese tramite la lettura di vari articoli pubblicati nelle piattaforme *Medium* [38] e *Statistics LibreTexts* [39].

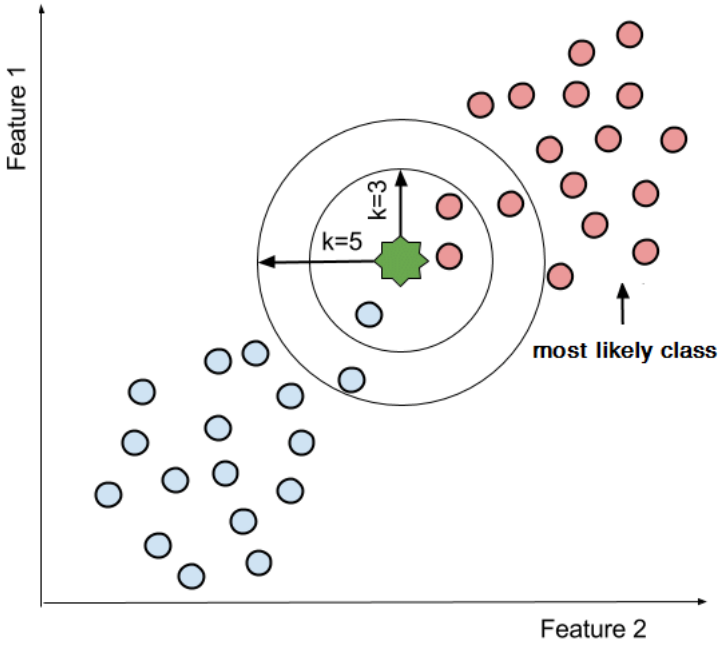


Figura 5.2: Esempio di funzionamento dell'algoritmo *K-Nearest Neighbors*

5.1.3 Naïve Bayes

I classificatori *Naïve Bayes* sono una famiglia di algoritmi probabilistici basati sull'applicazione del *Teorema di Bayes*.

Avendo un set di dati nella forma $\{X_1, \ldots, X_n, Y\}$, i generici X_j rappresentano le features del dataset e, il vettore colonna Y l'insieme delle label delle istanze del dataset. All'interno di Y sono contenuti diversi valori; tuttavia, l'insieme dei possibili valori univoci è costituito da $\{y_1, \ldots, y_k\}$.

Considerando X e Y come variabili casuali, il problema di classificazione può essere affrontato come un classico problema probabilistico. Infatti,

per identificare la classe di un campione, si deve determinare il valore di probabilità per tutti i possibili valori di $Y = y_1, \ldots, y_k$ tramite la formula:

$$P\left(Y = y_j \mid X = (x_1, \ldots, x_n)\right) \tag{5.4}$$

e, quindi, trovare il particolare valore y_j tale per cui il valore ottenuto con la 5.4 è massimo.

Tuttavia, il calcolo diretto della 5.4 risulta essere complicato; pertanto, si sfrutta il *Teorema di Bayes*:

$$P\left(Y = y_j \mid X = (x_1, \ldots, x_n)\right) = \frac{P\left(X = (x_1, \ldots, x_n) \mid Y = y_j\right) P\left(Y = y_j\right)}{P\left(X = (x_1, \ldots, x_n)\right)} \tag{5.5}$$

Gli elementi contenuti nella 5.5 prendono il nome di:

- Probabilità a posteriori: $P\left(Y \mid X\right)$;
- Probabilità a priori: $P\left(Y\right) = \frac{|Y = y_j|}{|\,Y\,|}$;
- Probabilità stato-condizionale: $P\left(X \mid Y\right)$;
- Probabilità dell'evidenza: $P\left(X\right)$. All'interno della 5.4, tale termine rimane uguale per tutti i possibili valori di $Y = y_j$. Di conseguenza, è possibile ignorarlo nel calcolo delle probabilità.

Tuttavia, nel caso di set di dati con numero elevato di features, identificare la combinazione $X = (x_1, \ldots, x_n)$ per il calcolo della probabilità stato-condizionale richiede tempi elevati. Pertanto, si assume che tutte le variabili del dataset sono *indipendenti* tra loro.

Malgrado la forte assunzione, in quanto nella realtà difficilmente si hanno variabili non correlate, la seguente relazione si semplifica notevolmente:

$$P\left(X = (x_1, \ldots, x_n) \mid Y = y_j\right) = P\left(X = x_1 \mid Y = y_j\right) \cdot \ldots \cdot P\left(X = x_n \mid Y = y_j\right) \tag{5.6}$$

A valle dell'ipotesi di indipendenza delle variabili, la 5.4 può essere riscritta nel seguente modo:

$$P\left(Y = y_j \mid X = (x_1, \ldots, x_n)\right) = \frac{P\left(Y = y_j\right) \prod_{i=1}^{n} P\left(x_i \mid Y = y_j\right)}{P\left(X = (x_1, \ldots, x_n)\right)} \tag{5.7}$$

Categorical Naïve Bayes

Nella sezione precedente, sono state introdotte le basi su cui poggiano gli algoritmi appartenenti alla famiglia Naïve Bayes. Lo scopo di quest'altra sezione è quello di descrivere il funzionamento dell'algoritmo che effettivamente è stato utilizzato e ha prodotto dei risultati ottimi, grazie al fatto che le feature considerate sono indipendenti fra di loro (vedi Analisi della correlazione lineare, Figura 4.8). Inoltre, le features fornite all'algoritmo soddisfano l'assunzione di *Distribuzione categorica*; di conseguenza, la probabilità che nell'i-esima feature x_i ci sia la categoria t data la classe c è data da:

$$P\left(x_i = t \mid y = c;\ \alpha\right) = \frac{N_{tic} + \alpha}{N_c + \alpha n_i} \tag{5.8}$$

Gli elementi contenuti nella 5.8 sono:

- N_{tic}: numero di volte che la categoria t con classe c è presente in x_i;
- N_c: numero totale di istante con classe c;
- α: parametro di appiattimento di Laplace, calcolato per poter gestire il problema della frequenza zero che altererebbe i risultati di probabilità;
- n_i: conteggio univoco di tutte le categorie.

Parte delle informazioni sui classificatori sono state reperite da un articolo online [40].

5.1.4 Altri algoritmi

Durante il processo di addestramento sono stati sfruttati ulteriori algoritmi di apprendimento automatico, appartenenti alla libreria *Scikit-learn*. Tali algoritmi sono stati successivamente scartati a causa delle basse performance:

- RandomForestClassifier;
- SVM (Support Vector Machine);
- GaussianNB;
- MultinomialNB;
- BernoulliNB;
- ComplementNB;
- MLPClassifier.

5.2 Algoritmi di apprendimento automatico non supervisionato

In questa sezione si descrivono i processi di addestramento e il funzionamento degli algoritmi di apprendimento automatico non supervisionato da un punto di vista teorico-procedurale-matematico.

5.2.1 K-means

K-means è un algoritmo di apprendimento automatico non supervisionato, usato per raggruppare insiemi di oggetti che presentano delle affinità. Il funzionamento di tale algoritmo si riassume tramite i passi indicati di seguito [41]:

- Fase di inizializzazione: occorre decidere il numero di k clusters da identificare;
- Passo 1: inizialmente l'algoritmo seleziona in maniera casuale k elementi dall'insieme degli oggetti. Tali elementi prendono il nome di centroidi (clusters iniziali);
- Passo 2: per ogni campione misura la distanza dai k centroidi. Il campione viene assegnato al cluster il cui centroide ha distanza minore;
- Passo 3: calcola la media di ogni cluster con la 5.9. La media viene considerata come nuovo centroide del cluster. Vengono ricalcolate tutte le distanze dei campioni dai nuovi centroidi e, se necessario, i campioni vengono riassegnati. Gli elementi contenuti nella formula: (i) M_k è il vettore delle medie; n_k è il numero di elementi del k-esimo cluster; x_{ik} è l'i-esimo elemento del k-esimo cluster;
- Passo 4: si reiterano i passi 1, 2 e 3 finché non viene soddisfatta una delle condizioni di arresto;
- Condizioni di terminazione: (i) per ogni cluster si calcola l'errore quadratico tramite la 5.10. L'algoritmo termina quando la somma dei k errori quadratici relativi ai k clusters raggiunge il suo minimo; (ii) gli elementi dei clusters si stabilizzano; (iii) si raggiunge un numero massimo di iterazioni.

$$M_k = \frac{1}{n_k} \sum_{i=1}^{n_k} x_{ik} \tag{5.9}$$

$$e_k^2 = \sum_{i=1}^{n_k} (x_{ik} - M_k)^2 \tag{5.10}$$

Capitolo 6

Risultati

6.1 Visualizzazione dataset

6.1.1 Risultati PCA

Il primo approccio alla visualizzazione del dataset è stato di tipo lineare: analisi delle componenti principali. Pur avendo fornito all'algoritmo dei dati scalati, dalla Figura 6.1 si evince che tale tecnica non restituisce nessuna informazione utile, se non che i preventivi approvati sono un tutt'uno con quelli non approvati.

Questo primo risultato potrebbe suggerire un'assenza di schemi ricorrenti tra i dati; ovvero, tutti i preventivi che vengono accettati, al contempo sono anche non accettati. In realtà questo risultato è dovuto al fatto che, per come sono distribuiti i dati, non esiste alcuna relazione di tipo lineare in grado di esprimere i dati sfruttando un nuovo sistema di coordinate che è una combinazione lineare dei campioni originali.

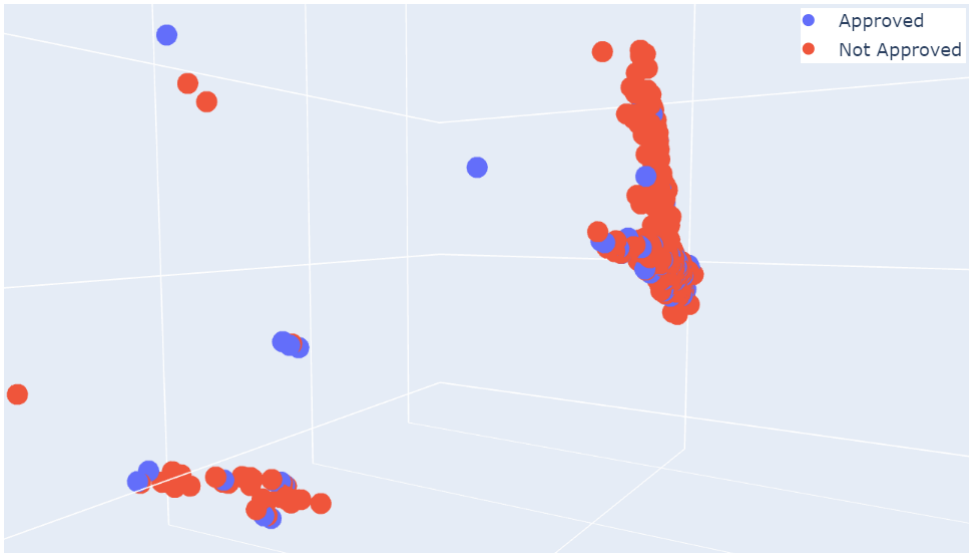


Figura 6.1: Visualizzazione del dataset tramite *PCA*

6.1.2 Risultati t-SNE

A valle del risultato della PCA, si è deciso di visualizzare il set di dati tramite tecniche di *embedding* non lineare. A tal proposito, si è fatto ricorso all'algoritmo t-SNE. I risultati derivanti dalla sua esecuzione sono rappresentati nella Figura 6.2, dalla quale si può osservare che si sono formati dei clusters di preventivi ben definiti. I preventivi approvati sono nettamente distinguibili da quelli non approvati. Tutto ciò fornisce una risposta anche alla domanda: "*perché i classificatori funzionano bene seppur in presenza di una dataset sbilanciato?*" (Figura 4.4).

In aggiunta, questa fase ha permesso di effettuare una considerazione di maggiore rilevanza rispetto alle precedenti. Osservando il grafico a dispersione, emerge un'ulteriore distinzione tra i preventivi: la *tipologia* di preventivo. In altre parole, non solo si è in grado di comprendere che sono presenti dei preventivi che non vengono mai commissionati, ma si riscontra che tra quelli che vengono sempre commissionati, a volte, *qualcosa* causa la perdita di potenziali lavori. Di conseguenza, dopo aver individuato i tipi di preventivi maggiormente commissionati e, condotto uno studio sui gap in tali contesti,

l'azienda può apportare dei miglioramenti mirati che provocano un effettivo accrescimento del processo produttivo.

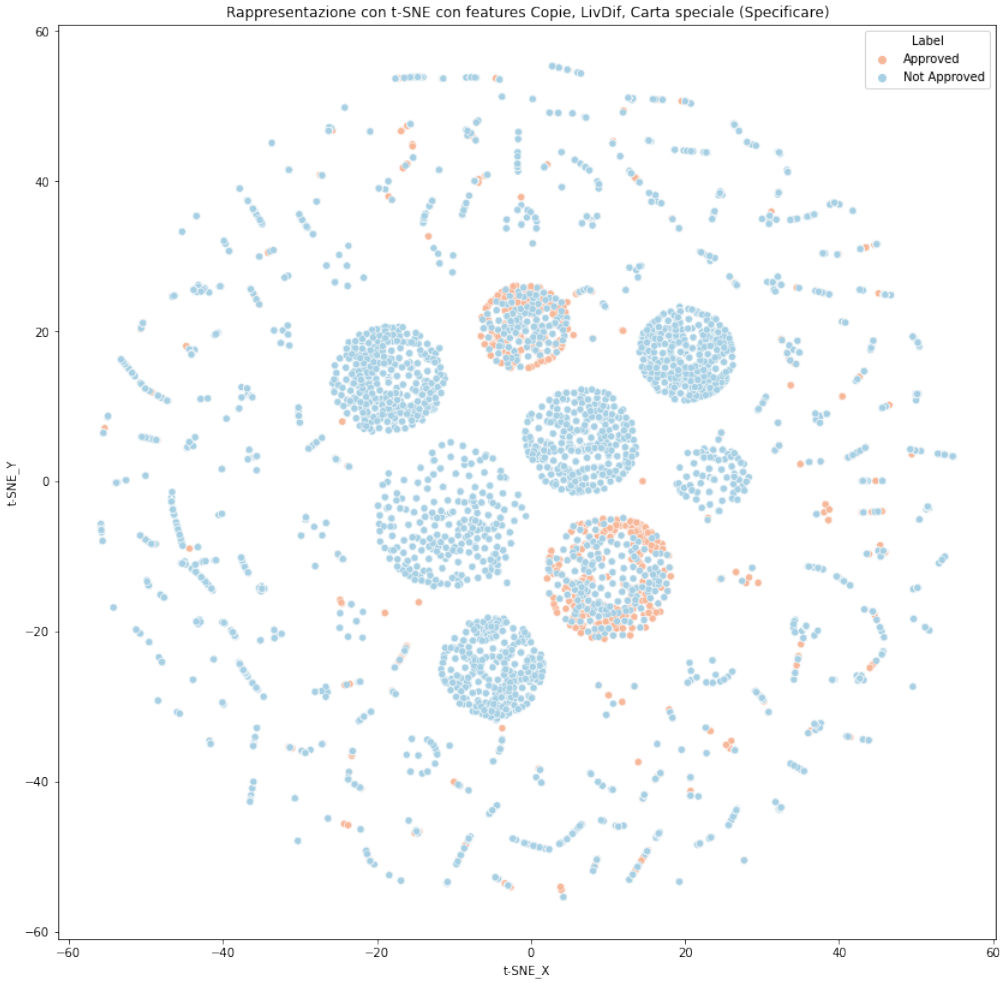


Figura 6.2: Visualizzazione del dataset tramite *t-SNE*

6.2 Metriche di valutazione

Terminato il processo di addestramento dei vari classificatori, si procede con la valutazione delle loro performance. A tal fine sono state utilizzate delle metriche molto comuni nel campo della *data-science* [42]:

- In generale, il dataset è stato suddiviso in training-set e test-set tramite tecniche di *train-test-split* e *cross-validation*. In entrambi i casi i livelli di accuracy ottenuti sono stati simili.

- Matrice di confusione: le righe contengono i valori reali, le colonne quelli predetti. Indice di un buon classificatore è una matrice diagonale. Questa matrice informa su quali sono le *classi più confuse*. In altre parole, comunica quanti esempi sono stati correttamente classificati.

- Accuracy: quantifica il tasso di predizioni corrette sul totale degli esempi. Tuttavia, tale parametro potrebbe essere fuorviante. Perciò, deve essere accompagnata da altre metriche;

$$A = \frac{TP + TN}{P + N} \tag{6.1}$$

- Precision: quantifica l'accuratezza con cui le classi positive vengono predette. La precision fornisce una misura utile in quanto avverte dell'eventuale crescita di falsi positivi.

$$P = \frac{TP}{TP + FP} \tag{6.2}$$

- Recall: quantifica il rapporto di istanze positive correttamente individuate. La recall fornisce informazioni importanti sull'eventuale crescita dei falsi negativi.

$$R = \frac{TP}{TP + FN} \tag{6.3}$$

- F1-score: è una media armonica ottenuta fondendo precision e recall;

$$F1 = \frac{2}{\frac{1}{R} + \frac{1}{P}} \tag{6.4}$$

- Curva ROC: nel caso di classificatore binario, la curva ROC si crea tracciando l'andamento dei valori del tasso dei veri positivi (*True Positive rate* – TPR) rispetto ai valori del tasso dei falsi positivi (*False Positive rate* – FPR). Se la curva ha un andamento $y = x$ (bisettrice del primo e terzo quadrante), allora le predizioni del classificatore sono completamente casuali. Altrimenti, un andamento asintotico tendente al TPR, è indice di buon classificatore.

  Una metrica associata alla curva ROC è l'AUC: *Area Under the Curve*. Il valore dell'area appartiene all'intervallo $[0, 1]$. Un valore tendente a 1 indica che le performance del modello sono eccellenti.

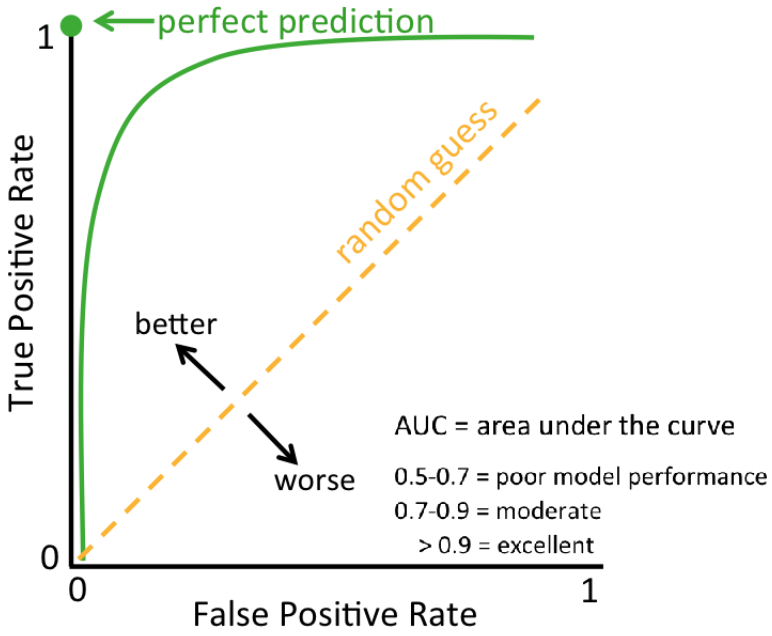


Figura 6.3: *Receiver Operating Characteristic*

6.3 Risultati Decision Tree Classifier

La semplicità e flessibilità sono i punti di forza di questo algoritmo. Dalle Figure 4.5 e 4.6 si osserva che le due classi di preventivo sono distribuite sullo stesso piano formando dei clusters. Il Decision Tree Classifier è stato in grado di percepire il pattern tra questi dati suddivisi in intervalli.[1]

6.3.1 DTC - dataset con tutte le features

In questa sezione sono stati inclusi gli esiti delle performance del Decision Tree Classifier (DTC) addestrato sul set di dati contenente tutte le features. Tali risultati sono relativi al test-set.

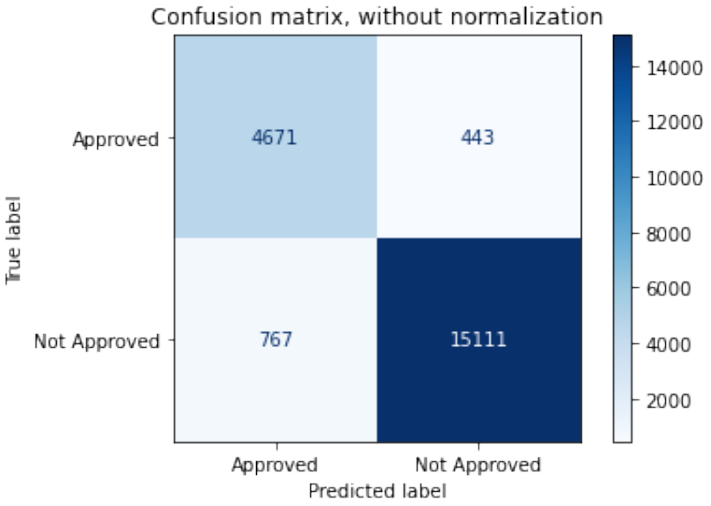


Figura 6.4: Matrice di confusione del test-set non normalizzata

[1] I risultati delle performance del classificatore sono identici sia nel caso che questo venga addestrato con training-set non bilanciato che bilanciato. Quindi, per comodità, gli esiti sono stati riportati una volta sola con duplice valenza.

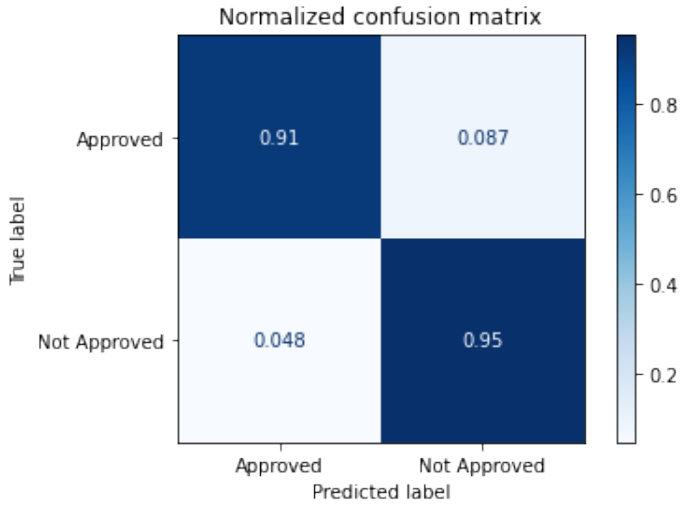


Figura 6.5: Matrice di confusione del test-set normalizzata

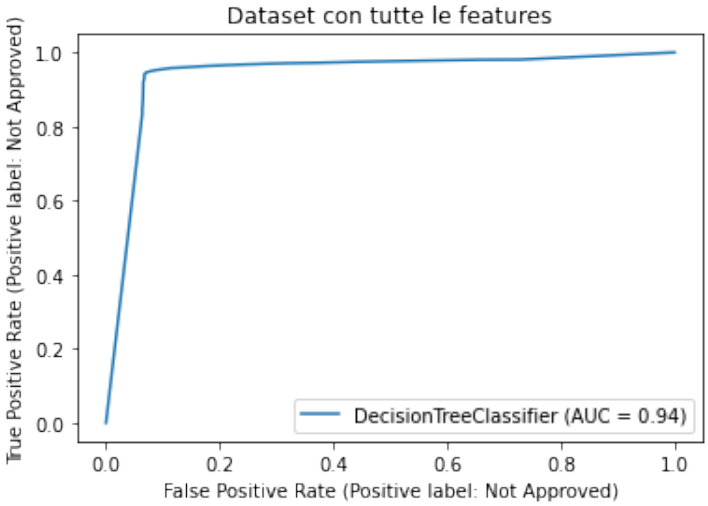


Figura 6.6: DTC addestrato sul training-set con tutte le features

	precision	recall	f1-score	support
Approved	0.86	0.91	0.89	5114
Not Approved	0.97	0.97	0.96	15878
Accuracy			0.94	20992
Macro avg	0.92	0.93	0.92	20992
Weighted avg	0.94	0.94	0.94	20992

Tabella 6.1: Tabella riassuntiva performance DTC – dataset originale

6.3.2 DTC - dataset ridotto

In questa sezione sono stati inclusi gli esiti delle performance del Decision Tree Classifier (DTC) addestrato sul set di dati ridotto a valle del processo di features selection. Tali risultati sono relativi al test-set.

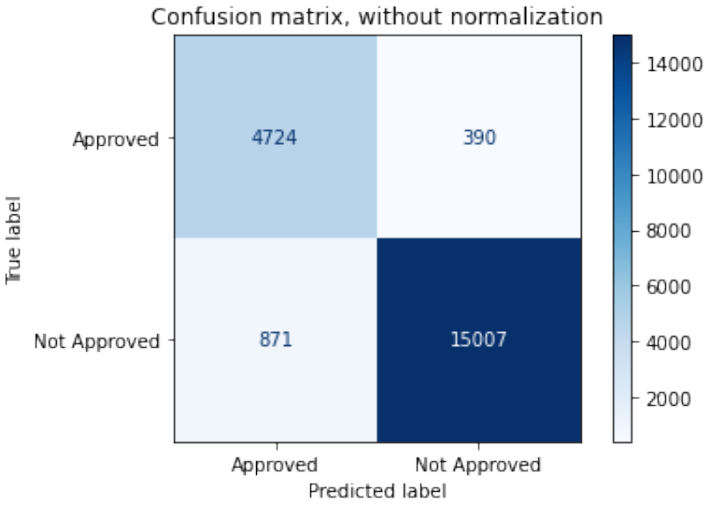


Figura 6.7: Matrice di confusione del test-set non normalizzata

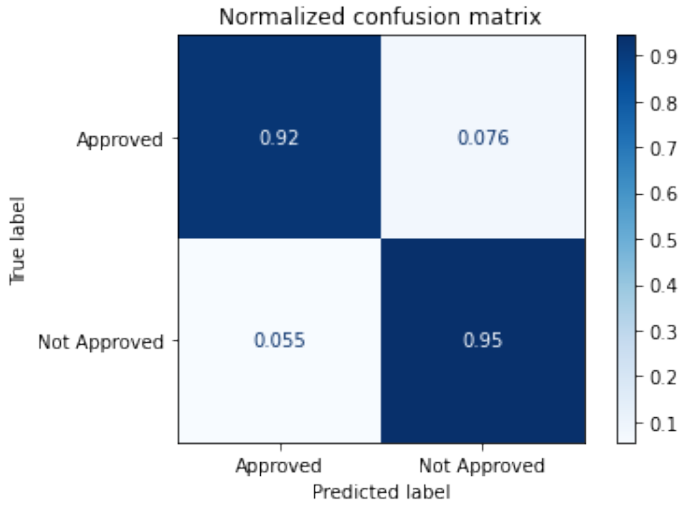


Figura 6.8: Matrice di confusione del test-set normalizzata

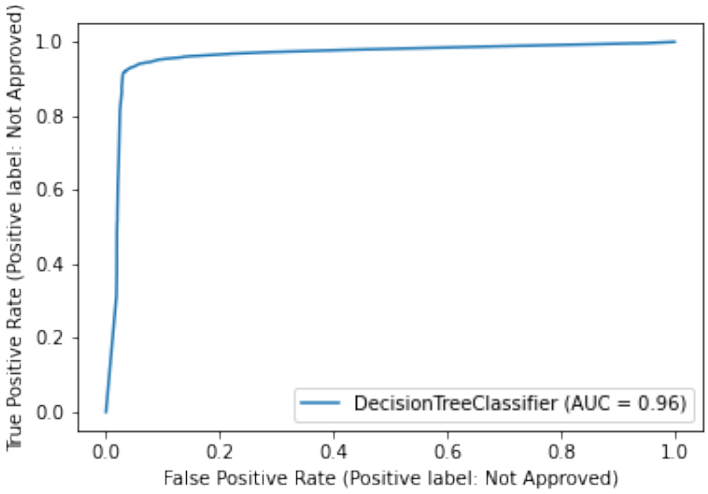


Figura 6.9: DTC addestrato sul training-set ridotto

	precision	recall	f1-score	support
Approved	0.84	0.92	0.88	5114
Not Approved	0.97	0.95	0.96	15878
Accuracy			0.94	20992
Macro avg	0.91	0.93	0.92	20992
Weighted avg	0.94	0.94	0.94	20992

Tabella 6.2: Tabella riassuntiva performance DTC – dataset ridotto

6.3.3 DTC - variabili linearmente dipendenti

In questa sezione è stato incluso il risultato delle performance del DTC in presenza di variabili linearmente dipendenti. Durante la fase di analisi della correlazione lineare, è stato osservato che le variabili numeriche 'CsTot' e 'Copie' sono linearmente dipendenti (Figura 4.9).

Sperimentalmente è stato possibile constatare che, l'eliminazione di una delle due variabili, ovvero di informazione ridondante, si riflette in un miglioramento delle prestazioni del classificatore.

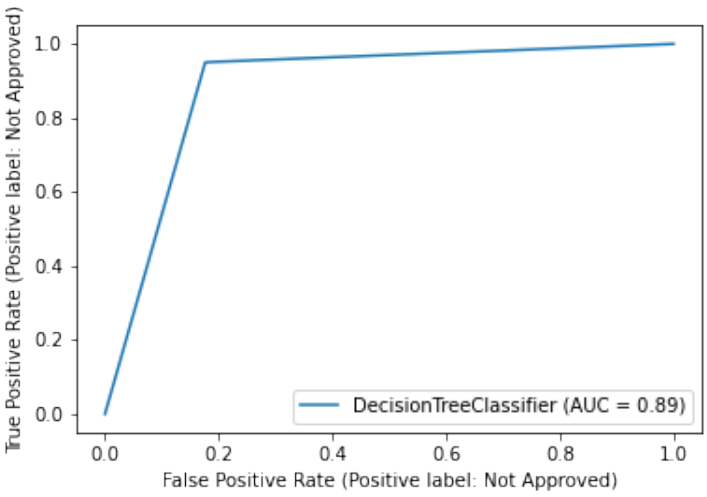


Figura 6.10: DTC con basse prestazioni

6.4 Risultati Catgorical Naïve Bayes

In questa sezione si presentano i risultati delle performance del classificatore Categorical Naïve Bayes (CNB) addestrato sul training-set ridotto. Gli esiti positivi, relativi al test-set, sono dovuti alla debole correlazione fra le variabili del set di dati.

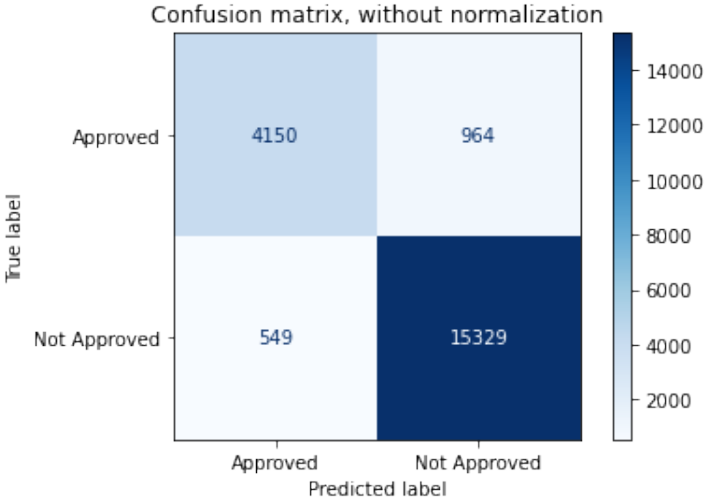


Figura 6.11: Matrice di confusione del test-set non normalizzata

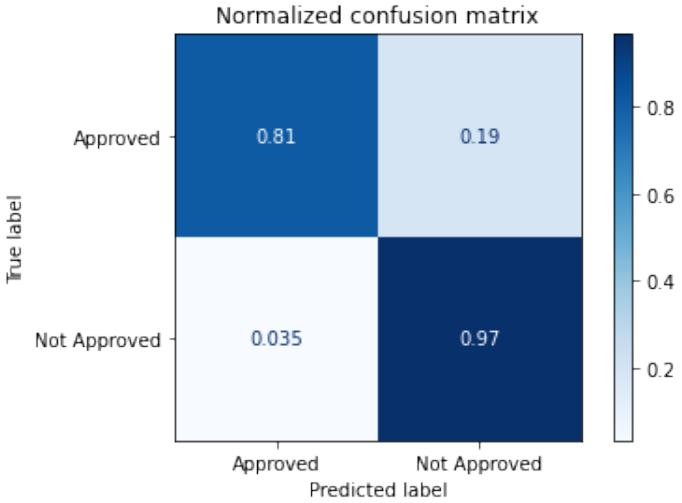


Figura 6.12: Matrice di confusione del test-set normalizzata

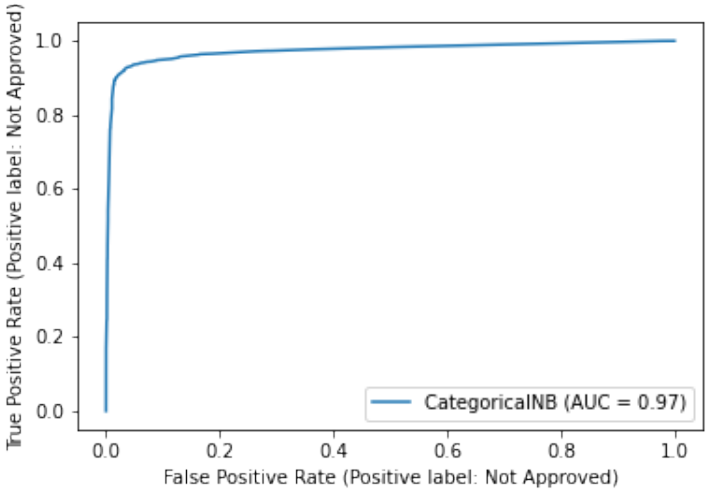


Figura 6.13: CNB addestrato sul training-set ridotto

	precision	recall	f1-score	support
Approved	0.88	0.81	0.85	5114
Not Approved	0.94	0.97	0.95	15878
Accuracy			0.93	20992
Macro avg	0.91	0.89	0.90	20992
Weighted avg	0.93	0.93	0.93	20992

Tabella 6.3: Tabella riassuntiva performance CNB – dataset ridotto

6.5 Risultati K-Nearest Neighbors

In questa sezione si espongono i risultati delle performance del classificatore K-Nearest Neighbors (KNN) addestrato sul training-set ridotto. Ancora una volta, la distribuzione a cluster dei dati permette a questo classificatore di cogliere facilmente gli schemi ricorrenti fra i dati.

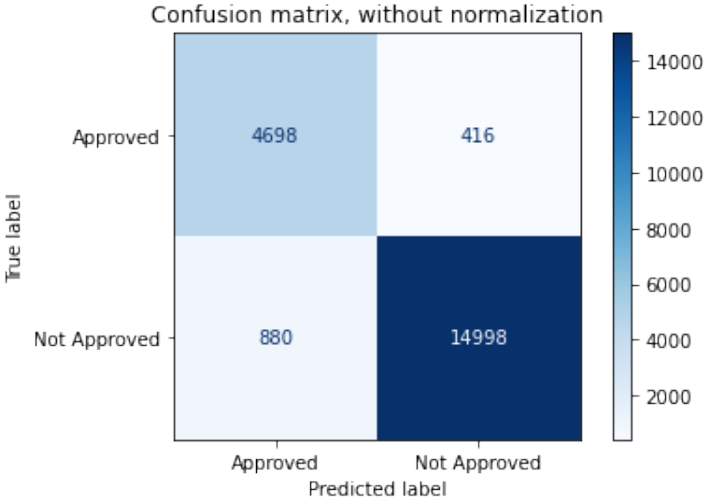


Figura 6.14: Matrice di confusione del test-set non normalizzata

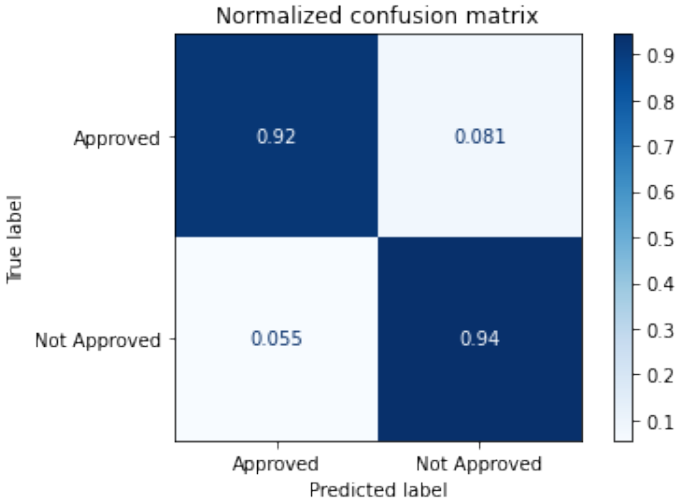


Figura 6.15: Matrice di confusione del test-set normalizzata

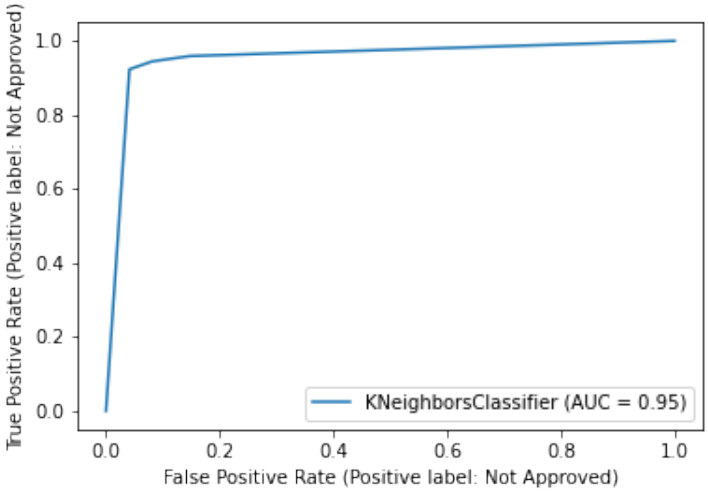


Figura 6.16: KNN addestrato sul training-set ridotto

	precision	recall	f1-score	support
Approved	0.84	0.92	0.88	5114
Not Approved	0.97	0.94	0.96	15878
Accuracy			0.94	20992
Macro avg	0.91	0.93	0.92	20992
Weighted avg	0.94	0.94	0.94	20992

Tabella 6.4: Tabella riassuntiva performance KNN – dataset ridotto

6.6 Risultati altri classificatori

Fatta eccezione per il RandomForestClassifier (scartato in quanto a parità di risultati risulta essere più complesso del DecisionTreeClassifier) gli altri sono dei pessimi classificatori. Infatti, il modello prodotto da un classificatore binario con un'accuracy pari a 0.5 fornisce predizioni totalmente casuali.

In aggiunta, non sono stati riscontrati miglioramenti neanche a seguito del processo di riduzione del numero di features.

6.6.1 Dataset con tutte le features

	Accuracy
RandomForestClassifier	0.927
LinearSVC	0.486
MultinomialNB	0.468
GaussianNB	0.378
BernoulliNB	0.755
ComplementNB	0.467
MLPClassifier	0.754

Tabella 6.5: Tabella riassuntiva performance – dataset originale

6.6.2 Dataset ridotto

	Accuracy
RandomForestClassifier	0.939
LinearSVC	0.273
MultinomialNB	0.596
GaussianNB	0.344
BernoulliNB	0.756
ComplementNB	0.481
MLPClassifier	0.498

Tabella 6.6: Tabella riassuntiva performance – dataset ridotto

6.7 Risultati K-means

In letteratura è noto che gli algoritmi di clustering lavorano meglio quando le distanze vengono scalate. Infatti, fornendo all'algoritmo il dataset non scalato, questo produce in output clusters sovrapposti. Di seguito si riportano i risultati ottenuti a seguito delle operazioni di scaling sui dati:

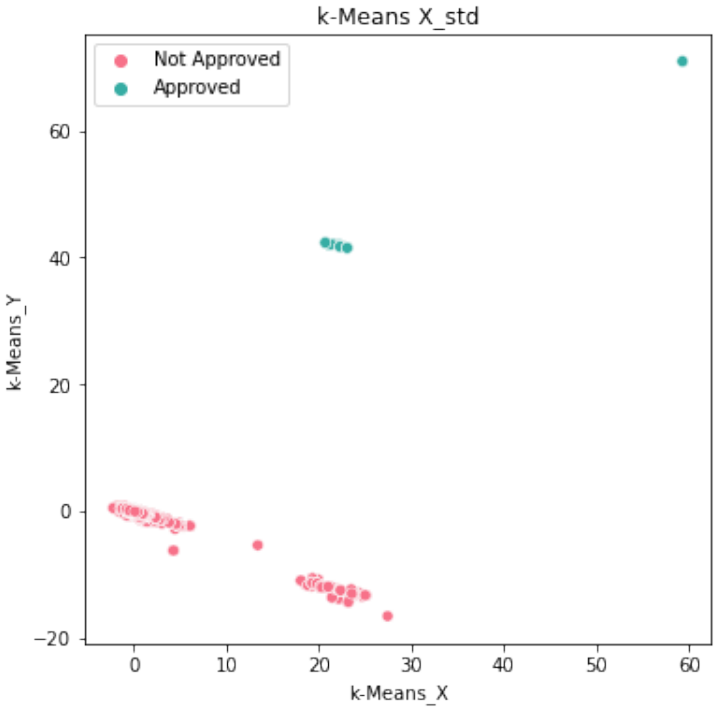


Figura 6.17: Clusters creati da K-means con il dataset standardizzato

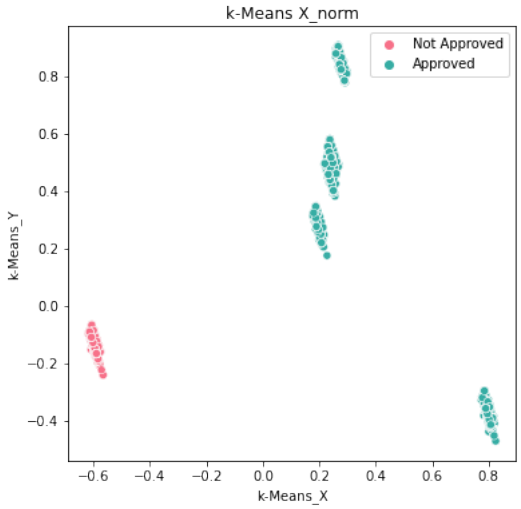


Figura 6.18: Clusters creati da K-means con il dataset normalizzato

6.8 Analisi statistica sulla feature Copie

Il processo di analisi discusso in questa tesi non è stato utile al solo fine di creare un sistema di diagnostica intelligente in grado di fornire predizioni sulle commissioni dei preventivi. Di fatto, è stato possibile comprendere che uno dei parametri decisivi in questo specifico caso studio è il numero di Copie del prodotto.

Di conseguenza, è stata condotta un'ulteriore analisi sulla distribuzione di frequenza delle copie dei preventivi approvati e non approvati, al fine di poter determinare quali sono i preventivi che vengono commissionati con maggiore e minore assiduità. L'azienda è quindi in grado di intervenire in maniera diretta e mirata sulla produzione.

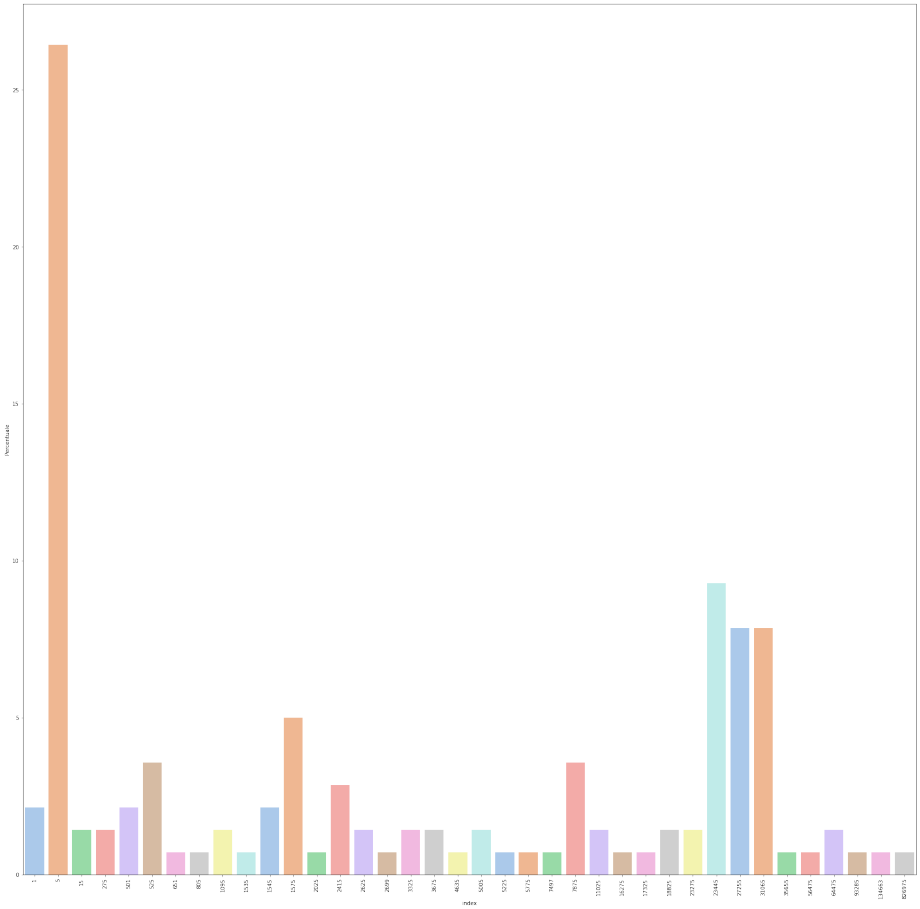


Figura 6.19: Frequenza della feature 'Copie' dei preventivi non approvati

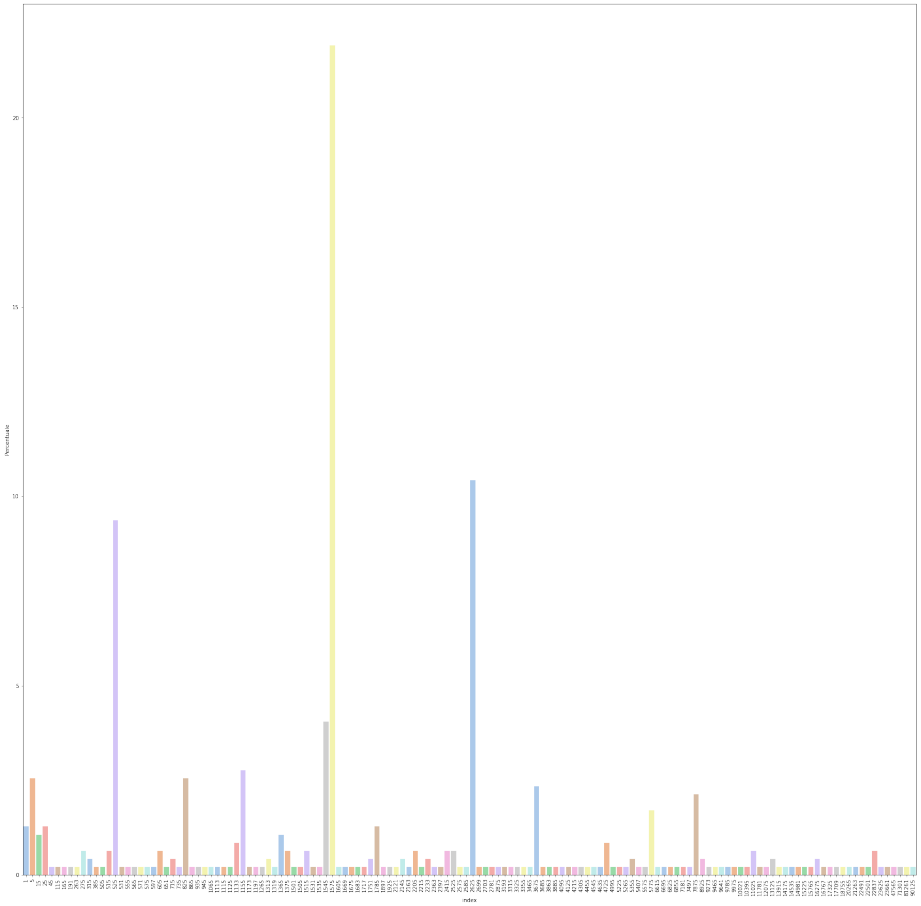


Figura 6.20: Frequenza della feature 'Copie' dei preventivi approvati

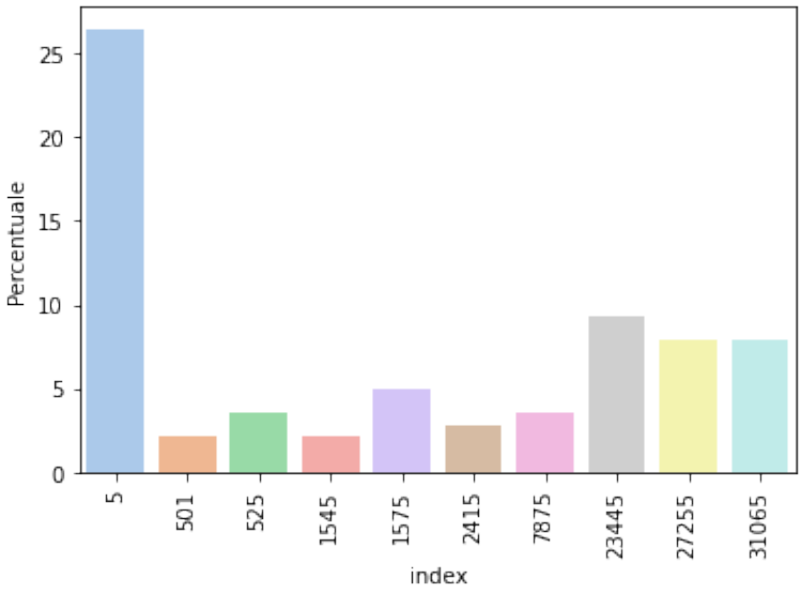


Figura 6.21: Dieci frequenze più ricorrenti della feature 'Copie' dei preventivi non approvati

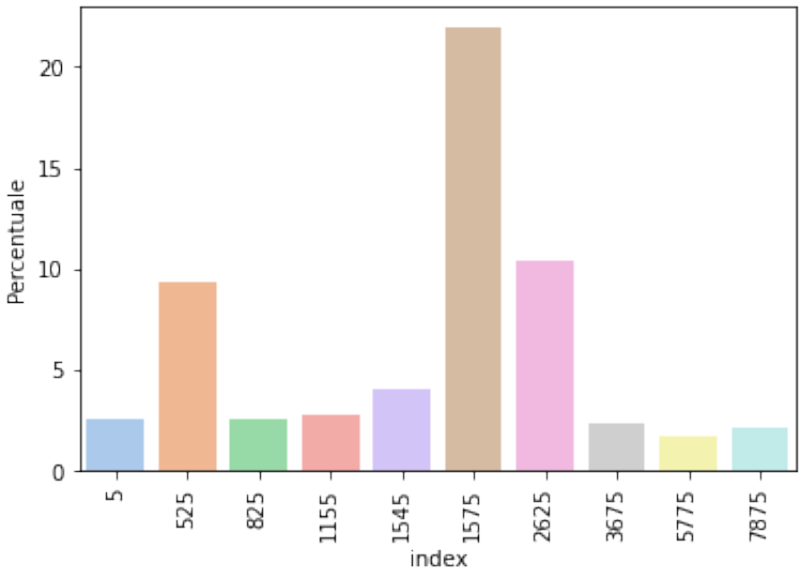


Figura 6.22: Dieci frequenze più ricorrenti della feature 'Copie' dei preventivi approvati

Conclusioni e sviluppi futuri

La produzione di questo settore ha una gestione su commessa, non su distinta base. Questo comporta che ogni prodotto deve essere stimato preventivamente e fabbricato sulla base delle specifiche del singolo prodotto.

Quando viene fatta la richiesta di un preventivo è fondamentale per l'azienda avere la consapevolezza del proprio valore competitivo su certe tipologie di prodotto, su certe fasce di produzione e in un contesto specifico.

Come anticipato in premessa, la struttura produttiva di cui si dota ogni azienda la porterà ad essere posizionata in un certo target di produzione.

Dall'analisi fatta, sono pertanto emersi una serie di elementi di valutazione che hanno fatto prevedere dei possibili sviluppi futuri del progetto riguardanti l'analisi dei preventivi, l'analisi della produzione e le scelte imprenditoriali.

Per quanto riguarda il singolo processo di preventivazione si sono ipotizzati due tipi di possibili applicazioni pratiche:

i. Servizio in cloud a pagamento che fornisce previsioni in relazione a: (i) dati che riceve tramite un form di una Web Application; (ii) GUI di un software proprietario;

ii. Software proprietario di realizzazione dei preventivi con inglobamento di un modulo intelligente in grado di fornire, in fase di redazione del preventivo, delle predizioni sulla commissione dei lavori.

Da un punto di vista più esteso, guardando verso il processo produttivo vero e proprio, l'aver individuato determinati elementi tecnici distintivi,

permette di esaminare la struttura produttiva stessa per analizzare eventuali problemi nei macchinari di produzione, per esempio in funzione del fatto che solo certe fasce di tiratura risultano competitive ed altre no o solo certi gradi di difficoltà del prodotto. Intervenire sulla struttura per colmare i *gap* strutturali e allargare la fascia di possibili offerte è ovviamente determinante per un'azienda di produzione.

L'applicazione dell'analisi nel progetto di studio è stata strutturata in un contesto di un prodotto specifico e di un'azienda specifica. L'estensione di tale ricerca ad altri contesti e ad altre realtà produttive determina un ampliamento della gamma di elementi distintivi, sia per altre tipologie di prodotto che di produzione.

Dotare l'azienda di un maggior numero di strumenti intelligenti di autodeterminazione consentirà di ridurre il margine di errore nell'immissione sul mercato. Pertanto, l'azienda potrà posizionarsi su un target ottimale o investire per acquisire maggiori competenze in altri ambiti.

Appendice A

Appendice

A.1 Matrice di covarianza e PCA

In questa sezione, si vuole dimostrare: "*Per determinare le componenti principali di un campione casuale multi-variato, è necessario calcolare gli autovalori e gli auto-vettori della matrice di covarianza associata al campione*" (in questo caso, il campione casuale multi-variato è il set di dati di partenza) [41].

Si vuole quindi espletare il legame che sussiste tra la risoluzione del problema di massimizzazione della varianza e gli autovettori della matrice di covarianza [43].

L'idea di fondo della PCA è che [44]:

- Ipotesi di *linearità*: si devono creare delle nuove componenti "riassuntive" che sono delle combinazioni lineari delle componenti della matrice di partenza X. Questa condizione si risolve ponendo: $Y = PX$.

  Y è la matrice che rappresenta i dati originali considerando gli autovettori di Σ_X come assi di riferimento, P è la matrice ortonormale di trasformazione che rappresenta un'applicazione lineare, X è il set di dati originale;

- Le sole componenti da considerare sono quelle con varianza massima (in quanto recano maggiore informazione), si tralasciano le informazioni ridondanti (variabili altamente correlate). Questa condizione si traduce prendendo in considerazione la matrice di covarianza Σ_X di X;
- Le componenti principali devono essere incorrelate. Questa condizione si traduce con l'ortogonalità dei vettori che formano le componenti principali. Analogamente, considerando la matrice di covarianza delle nuove coordinate dei punti nel nuovo sistema di riferimento, questa, deve essere una matrice diagonale.

Riassumendo, si può dire che l'idea è quella di trovare un nuovo sistema di riferimento che massimizza la varianza delle variabili. Il miglior sistema di riferimento che si può prendere in considerazione per rappresentare i dati di partenza è sicuramente formato dagli autovettori della matrice di covarianza. La motivazione viene spiegata tramite la dimostrazione che segue.

A.1.1 Strumenti di algebra lineare

Prima di addentrarci nella dimostrazione, è bene introdurre una serie di proprietà e notazioni – utili alla dimostrazione – apprese durante il corso di Geometria (Algebra Lineare) [45].

(a) $Y = PX$;

(b) Siano A e B matrici, $(AB)^T = B^T A^T$;

(c) Se A è una matrice qualsiasi, le matrici $A^T A = AA^T$ sono entrambe simmetriche;

(d) $\Sigma_X = \frac{1}{n-1} XX^T$ (è simmetrica per la (c); allora si può applicare il *Teorema spettrale* e il suo corollario);

(e) Sia $A \in M_{n\times n}(\mathbb{R})$. A è simmetrica se e solamente se esiste una matrice ortogonale E tale che $E^T AE$ è una matrice diagonale (Corollario *Teorema spettrale*);

(f) $P \equiv E^T$.

(g) Se P è una matrice ortogonale, allora $P^{-1} = P^T$;

(h) Sia $A \in M_{n\times n}(\mathbb{R})$ una matrice simmetrica. Allora esiste una matrice ortogonale P tale che $P^T AP$ è diagonale. Ovvero esiste una base ortonormale $\mathcal{B} = \{v_1, \ldots, v_n\}$ di $\mathbb{R}^n$ formata da autovettori di A (*Teorema spettrale*, la dimostrazione di questo teorema risolve una parte della dimostrazione della (e));

(i) Una matrice $A \in M_{n\times n}(\mathbb{R})$ si dice simmetrica se $A = A^T$ (Utile per capire la dimostrazione del corollario del *Teorema spettrale*);

(j) Al fine di non appesantire troppo la notazione, si ipotizza che X e Y sono matrici a media nulla, ovvero, matrici nella forma *Mean Deviation*. Infatti, sotto tale ipotesi, la matrice di covarianza è data da $\Sigma_A = \frac{1}{n-1}AA^T$; altrimenti $\Sigma_A = \frac{1}{n-1}\sum_{i=1}^{n}\left(A_i - \bar{A}\right)\left(A_i - \bar{A}\right)^\top$ ($\bar{A}$ è la media della colonna A_i)[1];

A.1.2 Dimostrazione

Inizialmente, è stato detto che il problema della PCA di una matrice di partenza X si risolve calcolando gli autovalori ed i relativi auto-vettori della sua matrice di covarianza Σ_X.

L'obiettivo da raggiungere è trovare una matrice ortonormale P dove $Y = PX$ tale che $\Sigma_Y = \frac{1}{n-1}YY^T$ è diagonale.

Imporre Σ_Y (matrice di covarianza di Y) è una matrice diagonale, implica che le variabili di $Y = \{y_1, \ldots, y_n\}$ hanno varianza massima e sono incorrelate tra loro.

[1]Si normalizza con $n-1$ e non con n poichè nel primo caso si ottiene uno stimatore *unbiased*, nell'altro no. "We divide by $n-1$ for the same reason as we do it when calculating sample standard deviations – it gives us a better estimator of the population equivalent." Buglear, J. (2013). Practical Statistics: A Handbook for Business Projects (pp. 57). India: Kogan Page.

Infatti, nella sezione iniziale, è stato detto che: le componenti da considerare sono quelle con varianza massima e si tralasciano le informazioni altamente correlate ovvero, ridondanti.

$$
\begin{aligned}
\Sigma_Y &= \frac{1}{n-1} YY^T \text{ per la (j)} \\
&= \frac{1}{n-1} (PX) (PX)^T \text{ per la (a)} \\
&= \frac{1}{n-1} (PX) X^T P^T \text{ per la (b)} \\
&= \frac{1}{n-1} P \left(XX^T\right) P^T \\
&= P\Sigma_X P^T \text{ per la (d) e la (j)} \\
&= P \left(P^T DP\right) P^T \text{ per la (d), (e) ed (f)} \\
&= \left(PP^T\right) D \left(PP^T\right) \\
&= \left(PP^{-1}\right) D \left(PP^{-1}\right) \text{ per la (g)} \\
&= Id_n DId_n \\
&= D
\end{aligned}
\tag{A.1}
$$

Cioè, è stato appena dimostrato che scegliendo opportunamente P si ottiene (una matrice con varianza massimizzata e ridondanza eliminata; ovvero una matrice diagonale):

$$\Sigma_Y = \frac{1}{n-1} YY^T = D \tag{A.2}$$

La maniera più opportuna per scegliere P è che sia una matrice ortonormale. Per ottenere P matrice ortonormale è necessario dunque, ottenere gli autovalori ed i relativi auto-vettori della matrice di covarianza Σ_X, in quanto per il *Teorema spettrale* si ha che: calcolando gli autovettori di Σ_X (matrice simmetrica), si ottiene una base ortonormale con cui poter creare la matrice ortonormale P.

Ecco spiegato il perché de: "*si dimostra che per determinare le componenti principali di un campione casuale multi-variato, è necessario calcolare gli autovalori e gli auto-vettori della matrice di covarianza associata al campione*".

A.2 Indice di Gini

Un generico set di dati si presenta nella forma: $\{A_1, \ldots, A_n, C\}$ con $C = \{c_1, \ldots, c_k\}$. Gli A_j, con $j \in [1, n]$, sono le n features del dataset. I c_i, con $i \in [1, k]$, sono le possibili k classi.[ii]

Per ogni feature A_j del dataset si calcola la somma pesata dei Gini: $Gini(A_j)$. Infine, si sceglie la feature con indice di Gini più basso.

Il set di possibili valori, contenuto all'interno di una generica feature A_j, si indica con $E = \{E_1, \ldots, E_h\}$. Quindi, il numero totale di istanze, è pari a $|E| = \sum_{i=1}^{h} |E_i|$; ovvero, la somma di tutte le cardinalità dei possibili valori determina il numero totale di istanze.

[ii]Esempio: FmPag, LivDif, Label con Label = {Approved, Not Approved}.

Il processo parte dal calcolo dell'indice di Gini dei valori $E_1, \ldots, E_h$ contenuti all'interno di A_j.

Tale indice è dato da $Gini\,(E_i) = 1 - \sum_{j=1}^{k} (p_j)^2$ con $i \in [1, h]$ e k numero totale delle classi.

Il valore $p_j = \frac{|\{t \in E_i : t[C] = c_j\}|}{|E_i|}$ rappresenta la porzione di elementi t contenuti in A_j che: assumono valore E_i e appartengono alla classe c_j sul totale degli esempi che assumono valore E_i indipendentemente dalla classe target.[iii]

Infine, la somma pesata degli indici Gini per ogni feature A_j è data da: $Gini\,(A_j) = \sum_{i=1}^{h} Gini\,(E_i)\,\frac{|E_i|}{|E|}$.[iv]

Parte delle informazioni riportate in questa sezione, sono state apprese tramite la lettura di vari articoli pubblicati nelle piattaforme *Medium* [36] e *Stack Exchange* [46].

[iii] Esempio: prendendo in considerazione la feature FmPag, i possibili valori che può assumenre sono: A, B. La porzione totale di elementi che assume valore A è 5, di cui 3 con target Approved e 2 con target Not Approved; la porzione totale di elementi che assume valore B è 6, di cui 2 con target Approved e 4 con target Not Approved;. L'indice Gini all'interno della colonna FmPag relativo al valore A è dato da: $Gini\,(A) = 1 - \left(\frac{3}{5}\right)^2 - \left(\frac{2}{5}\right)^2$ mentre, quello relativo a B: $Gini\,(B) = 1 - \left(\frac{2}{6}\right)^2 - \left(\frac{4}{6}\right)^2$.

[iv] Esempio: $Gini\,(FmPag) = Gini\,(A)\,\frac{5}{5+6} + Gini\,(B)\,\frac{6}{5+6}$.

Bibliografia

[1] H. Simon and M. Fassnacht. Machine learning in pricing. In *Price Management: Strategy, Analysis, Decision, Implementation*, chapter 9, page 342. Springer International Publishing, 2018.

[2] L. Columbus. How to drive more cpq sales with ai in 2020. https://www.forbes.com/sites/louiscolumbus/2020/02/06/how-to-drive-more-cpq-sales-with-ai-in-2020, Feb 2020.

[3] Sap cpq | soluzioni cpq (configure price quote). https://www.sap.com/italy/products/cpq.html.

[4] Oracle configure, price, quote. https://www.oracle.com/it/cx/sales/cpq.

[5] Ibm sterling configure, price, quote - ibm sterling configure, price, quote - panoramica. https://www.ibm.com/it-it/products/configure-price-quote.

[6] What is cpq, or configure, price, quote? https://www.salesforce.com/products/cpq/resources/what-is-cpq.

[7] Oracle configure, price, quote. https://www.oracle.com/it/cx/sales/cpq.

[8] Automate price quote requests. https://automationhero.ai/videos/automate-nlp-quote-requests, Oct 2020.

[9] T.M. Mitchell. *Machine Learning*. McGraw-Hill International Editions. McGraw-Hill, 1997.

[10] Applications for python. https://www.python.org/about/apps.

[11] Andrew Luashchuk. 8 reasons why python is good for artificial intelligence and machine learning. https://towardsdatascience.com/8-reasons-why-python-is-good-for-artificial-intelligence-and-machine-learning-4a23f6bed2e6, May 2019.

[12] What is numpy. https://numpy.org/doc/stable/user/whatisnumpy.

[13] Numpy. https://it.wikipedia.org/wiki/NumPy, Nov 2020.

[14] Pandas. https://pandas.pydata.org.

[15] Suhani S. What is pandas in python? everything you need to know. https://www.activestate.com/resources/quick-reads/what-is-pandas-in-python-everything-you-need-to-know, Mar 2021.

[16] Visualization with python. https://matplotlib.org.

[17] Seaborn: statistical data visualization. https://seaborn.pydata.org.

[18] Difference between matplotlib vs seaborn. https://www.geeksforgeeks.org/difference-between-matplotlib-vs-seaborn, Jul 2020.

[19] Scikit-learn. https://scikit-learn.org/stable.

[20] Project jupyter. https://jupyter.org.

[21] Introduction to sql. https://www.w3schools.com/sql/sql_intro.asp.

[22] Jason Brownlee. Why one-hot encode data in machine learning? https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning, Jun 2020.

[23] Alexandru Niculescu-Mizil, Claudia Perlich, Grzegorz Swirszcz, Vikas Sindhwani, Yan Liu, Prem Melville, Dong Wang, Jing Xiao, Jianying Hu, Moninder Singh, Wei Shang, and Yan Zhu. Winning the kdd cup orange challenge with ensemble selection. *Journal of Machine Learning Research - Proceedings Track*, 7:23–34, 01 2009.

[24] A. Rezzani. *Big Data Analytics. Il manuale del data scientist.* Apogeo education. Apogeo Education, 2017.

[25] Rahul Agarwal. The 5 feature selection algorithms every data scientist should know. `https://towardsdatascience.com/the-5-feature-selection-algorithms-every-data-scientist-need-to-know-3a6b566efd2`, Sep 2020.

[26] Renu Khandelwal. T-distributed stochastic neighbor embedding(t-sne). `https://towardsdatascience.com/t-distributed-stochastic-neighbor-embedding-t-sne-bb60ff109561`, Aug 2020.

[27] Difference between pca vs t-sne. `https://www.geeksforgeeks.org/difference-between-pca-vs-t-sne`, May 2020.

[28] Super User. Pca basata sulla matrice di covarianza. `https://www.webtutordimatematica.it/materie/statistica-e-probabilita/analisi-multivariata/analisi-delle-componenti-principali/pca-basata-sulla-matrice-di-covarianza`.

[29] Matrice delle covarianze. `https://it.wikipedia.org/wiki/Matrice_delle_covarianze`, Dec 2020.

[30] Laurens van der Maaten and Geoffrey Hinton. Viualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 11 2008.

[31] Kemal Erdem(burnpiro). t-sne clearly explained. `https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a`, Apr 2020.

[32] T-distributed stochastic neighbor embedding. https://it.wikipedia.org/wiki/T-distributed_stochastic_neighbor_embedding, Mar 2021.

[33] Divergenza di kullback-leibler. https://it.wikipedia.org/wiki/Divergenza_di_Kullback-Leibler, Jun 2020.

[34] Discesa del gradiente. https://it.wikipedia.org/wiki/Discesa_del_gradiente, Nov 2020.

[35] 1.10. decision trees. https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart.

[36] MLMath.io. Math behind decision tree algorithm. https://medium.com/@ankitnitjsr13/math-behind-decision-tree-algorithm-2aa398561d6d, Feb 2019.

[37] K. Koutroumbas and S. Theodoridis. The nearest neighbor rule. In *Pattern Recognition*, chapter 2, page 63. Elsevier Science, 2008.

[38] Ashwin Pandey. The math behind knn. https://ai.plainenglish.io/the-math-behind-knn-7883aa8e314c, Jan 2021.

[39] Libretexts. K-nearest neighbors (knn). https://stats.libretexts.org/@go/page/2483, Aug 2020.

[40] Reddy Suman Kumar. In depth naive bayes algorithm understanding. https://inblog.in/Categorical-Naive-Bayes-Classifier-implementation-in-Python-dAVqLWkf7E, Oct 2020.

[41] S. Ozdemir. *Data Science: guida ai principi e alle tecniche base della scienza dei dati.* Data Science. Feltrinelli Editore, 2017.

[42] Andrea Provino. Precision and recall con f1 score: Precisione e recupero - data science. https://andreaprovino.it/precision-and-recall-precisione-e-recupero, Nov 2020.

[43] Super User. Pca basata sulla matrice di covarianza. https://www.webtutordimatematica.it/materie/statistica-e-probabilita/analisi-multivariata/analisi-delle-componenti-principali/pca-basata-sulla-matrice-di-covarianza.

[44] Jonathon Shlens. A tutorial on principal component analysis derivation, discussion and singular value decomposition. 2003.

[45] Leonardo Biliotti. *Dispensa di geometria (algebra lineare).*

[46] Simone (https://stats.stackexchange.com/users/2719/simone). Mathematics behind classification and regression trees. Cross Validated. URL:https://stats.stackexchange.com/q/44404 (version: 2012-11-26).