LUISS

Corso di laurea in Marketing - Major in Analisi e Misure del marketing

Cattedra: Customer intelligence e logiche dei Big Data

Artificial intelligence in sustainable energy industry: prediction of customer churn rate

Marina Paolanti

RELATORE

Luca Romeo

CORRELATORE

750841

CANDIDATO

ANNO ACCADEMICO 2022/2023

Ringraziamenti

Vorrei dedicare questo spazio a chi, con dedizione e pazienza, ha contribuito alla realizzazione di questo elaborato.
Ringrazio il mio relatore Marina Paolanti, che in questi cinque mesi di lavoro, ha saputo guidarmi, con suggerimenti pratici, nelle ricerche e nella stesura dell'elaborato.
Grazie anche ai miei tutor Emanuele e Sofia per i loro preziosi consigli e per avermi suggerito puntualmente le giuste modifiche da apportare alla mia tesi.
Ringrazio di cuore i miei genitori, mia sorella e Giorgio. Senza i loro insegnamenti e senza il loro supporto, questo lavoro di tesi non esisterebbe nemmeno.
Ringrazio tutto lo staff dell'azienda Iren S.p.a., per il supporto economico e la fiducia nel devolvermi la borsa di studio che mi ha permesso di raggiungere questo importante traguardo.
Ed infine un ringraziamento particolare va alla mia Tutor aziendale Silvia e tutto il team commerciale di Iren Mercato per l'ospitalità e per le skills acquisite sul campo.

Indice:

1. Introduzione

Negli ultimi anni, l'avvento delle tecnologie informatiche ha trasformato il modo di fare Marketing e come le aziende gestiscono le informazioni sui loro clienti. La disponibilità di grandi volumi di dati sui clienti, resa possibile dai nuovi strumenti informatici, ha creato opportunità e sfide per le aziende che devono sfruttare i dati e ottenere un vantaggio competitivo.

Da un lato, molte organizzazioni si sono rese conto che le conoscenze contenute in questi enormi database sono fondamentali per supportare le varie decisioni organizzative. In particolare, le conoscenze sui clienti contenute in questi database è fondamentale per l'area Marketing all'interno delle aziende. Bisogna precisare però che molte di queste conoscenze utili sono nascoste e non sfruttate. D'altra parte, l'intensa concorrenza e l'aumento delle scelte disponibili per i clienti hanno creato nuove pressioni sui marketers e si è manifestata l'esigenza di gestire i clienti in una relazione a lungo termine. Questo nuovo fenomeno, chiamato *customer relationship management*, richiede che le organizzazioni adattino i loro prodotti e servizi e interagiscano con i propri clienti in base alle preferenze effettive dei clienti, piuttosto che su un presupposto generale di caratteristiche generali presunte. È sempre più diffusa la consapevolezza che un'efficace gestione delle relazioni con i clienti può essere realizzata solo con una vera comprensione delle esigenze e delle preferenze dei clienti stessi. In queste condizioni, gli strumenti di data mining possono aiutare a scoprire la conoscenza nascosta e a comprendere meglio i clienti, mentre un sistematico sforzo di gestione della conoscenza può incanalare la conoscenza in strategie di Marketing più efficaci. Questo rende lo studio dell'estrazione e della gestione della conoscenza particolarmente utile per il Marketing.

In questo studio, in particolare, vengono delineate le tecniche di machine learning necessarie per prevedere il tasso di abbandono dei clienti di un provider energetico neozelandese. E, più in generale, viene dimostrato come gli strumenti di Data e Analytics possono essere molto utili per le aziende per analizzare il tasso di abbandono dei propri clienti e per creare strategie di Marketing efficaci in grado di minimizzare il più possibile il numero di churners.

L'analisi dimostra, che per gestire e ridurre il tasso di abbandono è necessario adottare un approccio predittivo basato su dati, algoritmi statistici e tecniche di machine learning per fare previsioni basate sui dati storici.

Molti studi dimostrano che spesso la previsione del tasso di abbandono viene effettuata tramite problemi di classificazione; in questo studio infatti viene applicato un task di classificazione binario. La classificazione ha come obiettivo la costruzione di un modello che sia in grado di prevedere l'appartenenza di un'osservazione ad una specifica classe target, che può essere multinomiale oppure binaria, proprio come nel caso della classe target qui considerata "churn". Nell'approccio supervisionato, trattato nel corso di questa sperimentazione, il classificatore generalizza ed apprende la classificazione a partire da esempi per cui è esplicitato il corretto output.

La struttura dell'elaborato si articola come segue: nella fase introduttiva verrà approfondito il tema del tasso di abbandono e del ruolo che esso riveste all'interno delle aziende. Successivamente verrà illustrata la fase di sperimentazione in cui vengono presentate le varie fasi di modellazione da seguire per ottenere una buona previsione del churn del cliente. Essa si articola principalmente in tre sezioni; nella prima sezione denominata "Stato dell'arte" sarà fornita un'analisi della letteratura relativa agli studi precedentemente svolti in questo contesto. Nella sezione "Materiale e metodi" sarà descritto il dataset oggetto della sperimentazione e gli strumenti utilizzati ai fini della computazione stessa, verranno introdotti i modelli di classificazione costruiti e le misure di prestazione; nella sezione "Risultati e discussioni" verrà fornita un'illustrazione e una discussione dei risultati ottenuti. Infine, nella sezione "Conlusioni e Sviluppi futuri" si suggeriranno i potenziali sviluppi futuri di questo lavoro.

2. Il Churn Rate:

Il *churn rate*, anche detto tasso di abbandono o tasso di defezione, esprime la percentuale di clienti che ha abbandonato un servizio in un dato periodo di tempo rispetto al numero totale di clienti che ne ha usufruito nello stesso periodo *(Cocuzza D., Glossariomarketing.it*). Il churn rate è una metrica essenziale per capire lo stato di salute di un'impresa e le sue prospettive future; essa, infatti, fa parte dei principali indicatori di Marketing, fondamentali per comprendere l'andamento di una attività. I KPI Marketing (*Key Performance Indicator*) sono quelle metriche, o indicatori, che servono a misurare i risultati oggettivi, ovvero l'efficacia e l'efficienza, delle iniziative di attrazione e conquista di un cliente, l'acquisizione di contatti, il livello di soddisfazione e ingaggio, la qualità dell'esperienza, il ritorno degli investimenti nella produzione e promozione dei contenuti, ed effettuare quindi un'analisi quantitativa del ROI del budget allocato *(Della Bella F., www.digital4.biz*). La rapida evoluzione del marketing verso il digitale e le *MarTech* fa sì che le metriche utilizzate non siano sempre le stesse. Uno studio di *Salesforce* realizzato nel 2021 intervistando i Marketing manager in tutto il mondo mostra come i KPI utilizzati siano cambiati da un anno all'altro. Il 78% delle aziende dichiara di aver cambiato le proprie metriche, puntando di più sulla soddisfazione del cliente, sul Customer acquisition costs (CAC), e sul tasso di abbandono dei clienti stessi. Si assiste ad una vera e propria transizione da un approccio di tipo *product centric* a un approccio di tipo *customer centric* con una crescente consapevolezza della necessità di aumentare l'attenzione verso i fattori legati al cliente, come la soddisfazione del cliente, il servizio al cliente, la fidelizzazione del cliente e la qualità percepita dal cliente (*Shah, D., Rust, R., et al., 2006)*.

In questo contesto, quindi, il churn rate riveste un ruolo fondamentale poichè permette di quantificare il numero di clienti non più soddisfatti del servizio o prodotto offerto spesso a favore di altre imprese più competitive.

Il concetto di churn è legato a filo doppio con il fenomeno della liberalizzazione del mercato: nel momento in cui il cliente ha la possibilità di ottenere il medesimo servizio da più di una società, infatti, esso naturalmente opterà per quello che si rende più competitivo sul piano del trade-off qualità - prezzo.

Il churn rate, quindi, è una metrica essenziale per valutare le performance di marketing del proprio business, ma non solo; questo valore può essere utilizzato anche per una serie di considerazioni aggiuntive, essenziali per curare la performance aziendale.

Ad esempio, per analizzare l'andamento del processo di fidelizzazione dei propri clienti. Può aiutare a identificare l'impatto di eventuali cambiamenti nella propria offerta, nonché a calcolare il customer lifetime value e, infine, permette di fare previsioni sulle performance future della propria attività. Comprendere il tasso di abbandono dei clienti è essenziale per valutare l'efficacia dei propri sforzi di marketing e la soddisfazione generale dei clienti.

Il churn rate è inversamente proporzionale al retention rate: quanto più basso è il churn rate, tanto più alto è il retention rate e viceversa. Incrementare il tasso di fedeltà della clientela, minimizzando il tasso di abbandono, è l'obiettivo primario dell'impresa orientata al marketing. A tal fine, l'impresa si avvale solitamente dell'impiego di attività di marketing e promozione: essa, ad esempio, può ricorrere ad incentivi e sconti per indurre il cliente alla ripetizione dell'acquisto, o alla prova di nuovi prodotti. Migliorare il tasso di abbandono, quindi, può avere un impatto estremamente benefico sul business: secondo uno studio di *Bain&Co* e *Harvard Business Review*, una riduzione del tasso di abbandono del 5 percento porta a un incremento dei profitti dell'azienda tra il 25 e il 95 percento *(Gallo A., hbr.org).*

Per monitorare l'andamento di questa metrica e delineare strategie di marketing efficaci le aziende si avvalgono della cosiddetta *churn analysis*. La *churn analysis* è un'analisi previsionale che consente di individuare i clienti che presentano una maggiore probabilità di passare alla concorrenza, al fine di intervenire in anticipo ed evitarne la migrazione. Grazie all'analisi delle transazioni e delle altre informazioni disponibili sulla clientela elaborate con l'ausilio di sistemi di CRM, l'impresa può monitorare il grado di soddisfazione del cliente o *customer satisfaction* e pianificare azioni dirette ad aumentarlo, nell'intento di evitare che i propri clienti la abbandonino.

Un diagramma di flusso per l'elaborazione dei dati di abbandono della clientela è mostrato nella figura 1.

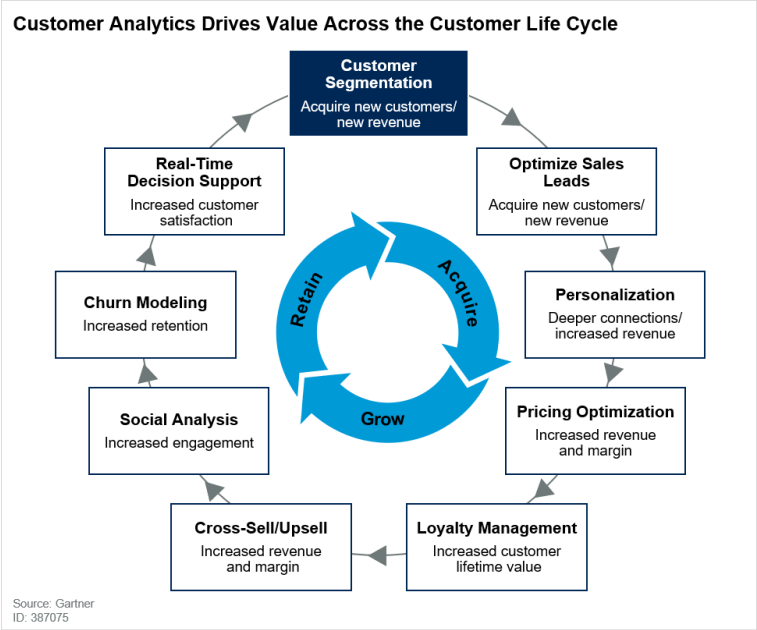


Figura 1: Customer Analytics: Uses, Features, & Tools in 2021 – Indicative

Le principali cause di abbandono:

L'analisi delle cause che spingono i clienti all'abbandono è una fase cruciale per i marketers poiché permette di capire come intervenire per massimizzare la soddisfazione dei clienti e ridurre il tasso di abbandono.

Le cause che spingono i clienti ad abbandonare un servizio vengono suddivise principalmente in quattro categorie diverse *(Len Markidan, groovehq.com):*

- Servizio clienti scadente: Il cliente può decidere di rescindere dal contratto nel caso in cui ritenga che l'azienda non abbia prestato accurata assistenza o se addirittura si sente trascurato: uno studio effettuato da Oracle, infatti, attesta che 9 clienti su 10 abbandonano il servizio stipulato con una società a causa di una cattiva esperienza (*Oracle, 2011*).

- Cattiva gestione dell'onboarding: Il termine *onboarding***,** che significa letteralmente "salire a bordo"- si riferisce al processo attraverso il quale un'azienda introduce un nuovo cliente al proprio prodotto o servizio. L'obiettivo principale è quello di assicurarsi che il cliente comprenda come utilizzare il prodotto o il servizio e che si senta a proprio agio nell'utilizzarlo.

  Tale processo può includere la formazione, il supporto, la configurazione del prodotto ed altre attività pensate per aiutare il cliente ad ottenere il massimo valore dal prodotto in questione. L'onboarding consente al cliente di comprendere come il prodotto o servizio può essere utile a raggiungere i suoi obiettivi *(Soltani, R., Nguyen, et.al., 2018).* Spesso, infatti, accade che la società ritenga un cliente acquisito quando per la prima volta esso sottoscriva ad un servizio, non tenendo tuttavia conto che, nel tempo che intercorre tra questa prima fase e il momento in cui per la prima volta il cliente realizza la propria soddisfazione, esso può facilmente stancarsi e abbandonare.

- Mancata realizzazione delle aspettative del cliente: Anche nel caso in cui l'azienda porti a buon fine il processo di *onboarding*, al fine di preservare il cliente essa deve costantemente porre attenzione alla sua soddisfazione. Infatti, nel caso in cui si sentisse trascurato, o più specificatamente ritenesse di non trarre più, o di non aver ancora tratto dopo un consistente lasso di tempo, alcun vantaggio dalla affiliazione alla società o al servizio, si incorrerebbe in un caso di abbandono a favore del competitor che garantisse un vantaggio maggiore.

- Cause naturali: Esistono una serie di fattori parzialmente indipendenti dalla volontà dell'azienda che possono portare all'abbandono del cliente aziendale. Si pensi al caso in cui il cliente non avesse più bisogno del servizio fornito dall'azienda, oppure all'eventualità della vendita della casa e del conseguente distacco delle utenze. In realtà, anche in questi casi, se l'azienda dovesse aver lasciato il cliente completamente soddisfatto, non incorrerebbe in alcuna perdita per l'azienda: il cliente potrebbe sì andarsene, ma per esempio consigliando il *brand* a conoscenti, oppure sottoscrivendo un nuovo contratto nella casa nuova.

2.1 Perchè la customer retention è importante per i profitti di un'azienda:

Molte aziende concentrano i propri sforzi di crescita verso l'acquisizione di nuovi clienti. Si tratta di un'attività essenziale in termini di crescita e sopravvivenza di un business ma allo stesso tempo le aziende sottovalutano l'impegno a "trattenere" i clienti già acquisiti, in particolare quelli che generano un ritorno di valore maggiore. Favorire il riacquisto risulta meno dispendioso e più remunerativo rispetto all'acquisizione di nuovi clienti. Inoltre, poichè l'economia è diventata sempre più orientata ai servizi l'importanza di mantenere le relazioni con i clienti oggi è più critica che mai (King, G. J., Chao, X., & Duenyas, 2016). Per questo motivo è necessario introdurre il concetto di customer retention.

La *customer retention* rappresenta quell'insieme di azioni che un business compie per trattenere i clienti già acquisiti, riducendone al minimo l'abbandono; si fa riferimento ad un vero e proprio processo di fidelizzazione del cliente *(Aspinall, E., Nancarrow, C., & Stone, M.,2001).* La customer retention è legata alla soddisfazione del cliente, affinché questa ci sia è necessario che il prodotto o servizio offerto sia di qualità o percepito come tale. L'obiettivo finale della fidelizzazione dei clienti è quello di massimizzare il *lifetime value* e la redditività di ciascun cliente.

Le statistiche dimostrano che un aumento del 5% della fidelizzazione dei clienti può far crescere i profitti di un'azienda dal 25% a circa il 95% in un determinato periodo di tempo (*Reichheld F., Schefter P., hbswk.hbs.edu).*

Di seguito la formula per calcolare il customer retention rate:

$$Customer\ retention\ rate = \frac{Customers\ at\ end\ of\ period - Customers\ acquired\ during\ period}{Customers\ at\ the\ start\ of\ the\ period}\ x\ 100$$

Il livello di fidelizzazione dei clienti varia a seconda della fase del ciclo di vita dell'azienda. Ad esempio, un'azienda nuova, priva ancora di clienti attuali, si concentrerà prettamente sulla

strategia di acquisizione, ma quando inizierà a guadagnare trazione e a conquistare clienti, dovrà ottimizzare la propria strategia e il proprio budget verso la fidelizzazione. Come mostrato nel grafico sottostante, man mano che ci si sposta verso destra, e quindi all'aumentare del ciclo di vita dell'organizzazione si aumenterà il processo di ritenzione dei clienti. È importante però precisare che le due attività di acquisizione e ritenzione devono essere esercitate in parallelo evitando di ignorare l'una o l'altra. Si tratta quindi di trovare un equilibrio tra le due. La scelta di concentrarsi maggiormente sull'acquisizione o sulla fidelizzazione dei clienti è fortemente influenzata dal punto in cui si trova l'organizzazione nel suo ciclo di vita.

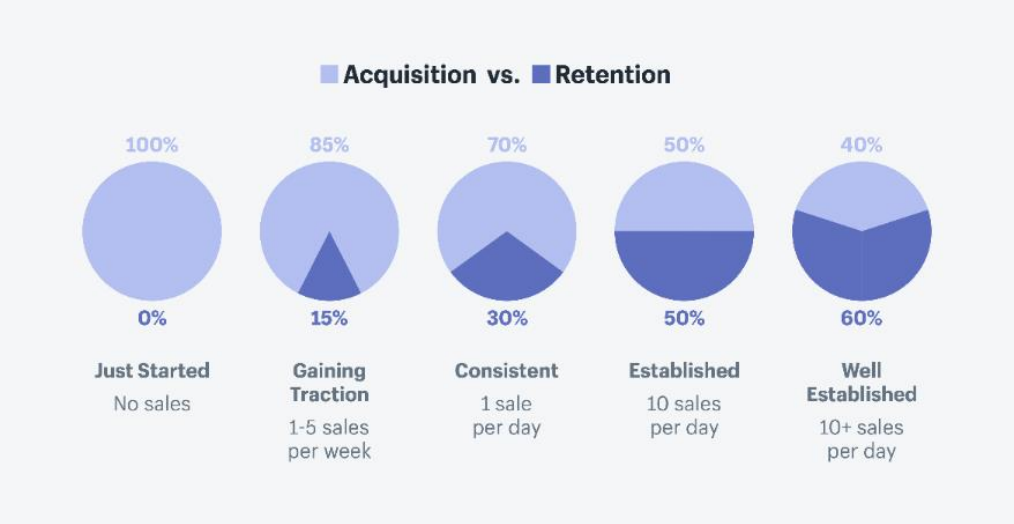


Figura 2: shopify.com

Esistono principalmente 3 ragioni per cui la fidelizzazione è un fattore importante per qualsiasi azienda che abbia clienti:

1. La retention permette di risparmiare denaro: l'acquisizione dei clienti è molto costosa e competitiva, concentrando tempo e sforzi sulla strategia di fidelizzazione dei clienti il budget viene allocato in maniera più efficace ed efficiente.
2. La retention aumenta la redditività: come affermato dagli studiosi Reichheld F. e Schefter P., è ormai noto che anche un aumento del 5% dei tassi di fidelizzazione può aumentare la redditività dell'azienda dal 25% al 95% *(Reichheld, F. F., Schefter, P., 2000).*

   Ci sono diversi motivi per cui i clienti che riacquistano aumentano la redditività:

- I clienti che ritornano sono da 3 a 10 volte più propensi a ripetere gli acquisti grazie alla fiducia che si è creata, cioè se la loro precedente esperienza con un determinato brand marchio è stata positiva.
- Il costo medio per spesa di un cliente di ritorno aumenterà nel corso della sua vita.
- I clienti fedeli hanno maggiori probabilità di rispondere bene all'upselling e al cross-selling.

3. La retention favorisce le referenze: Un aspetto fondamentale delle attività di fidelizzazione consiste nella creazione di un'esperienza positiva e piacevole per i clienti.

Infine, secondo uno studio di Forbes le principali tecniche per generare profitto da parte dei clienti fidelizzati sono due; Cross-Selling o Upselling ai clienti fedeli e minori risorse coinvolte (kumar S., www.forbes.com):

- *Cross-Selling* o *Upselling* ai clienti fedeli: Dal momento che i clienti fedeli conoscono meglio il marchio ci sono molte opportunità per fare upselling di prodotti; per up selling si intende la tecnica con la quale vengono offerti al cliente prodotti o servizi di maggiore qualità rispetto alla loro prima scelta. Alcuni studi dimostrano che i clienti esistenti hanno il 50% di probabilità in più di provare nuovi prodotti e spendono il 31% in più rispetto ai nuovi clienti (*Rioux P., forbes.com).*

- Minori risorse coinvolte: Per fidelizzare un cliente sono necessarie minori risorse. La fidelizzazione dei clienti richiede un reparto di customer success che si occupi di controllare periodicamente i clienti e un team di assistenza che risolva i problemi o i bug periodici. Spesso si ricorre anche all'uso di incentivi come sconti, piccoli regali o buoni. Tutte queste misure sono complessivamente più economiche rispetto a quelle che prevedono di spendere il budget in pubblicità e di utilizzare le risorse e il tempo per acquisire un nuovo cliente.

2.2 Il Customer relationship management:

Uno degli approcci più comunemente usati dalle aziende per ridurre il tasso di abbandono è il Customer relationship management.

Il *Customer relationship management* (CRM) è l'insieme di processi e attività d'impresa diretti alla gestione delle relazioni con il portafoglio clienti al fine di acquisire, mantenere e accrescerne il valore *(Kumar, V., Reinartz, W., 2018).* È una strategia di business o una filosofia aziendale il cui obiettivo è stabilire relazioni forti e durature con i clienti attuali e potenziali sulla base dell'analisi di informazioni che li riguardano. Attraverso lo studio di tali informazioni, infatti, l'impresa è in grado di monitorare il livello di soddisfazione del cliente e facilitarne il processo di fidelizzazione .

La funzione del CRM è quindi quella di individuare la strategia e gli strumenti da utilizzare per migliorare la capacità dell'azienda di comprendere le abitudini e i bisogni del cliente, così che l'azienda stessa possa avere con essi il miglior rapporto possibile e possa fornirgli gli strumenti giusti per relazionarsi con essa. Tutto questo è possibile grazie all'utilizzo di database in cui vengono concentrate tutte le informazioni riguardanti la clientela: le preferenze d'acquisto, gli ordini abituali, i dati demografici e le informazioni di contatto *(Francis, B., Ornati, M., 2012).*

Il CRM, quindi, è uno strumento ormai indispensabile nell'economia moderna che consente alle aziende non solo di accrescere la soddisfazione dei propri clienti, ma anche di aumentare la produttività e ridurre i costi.

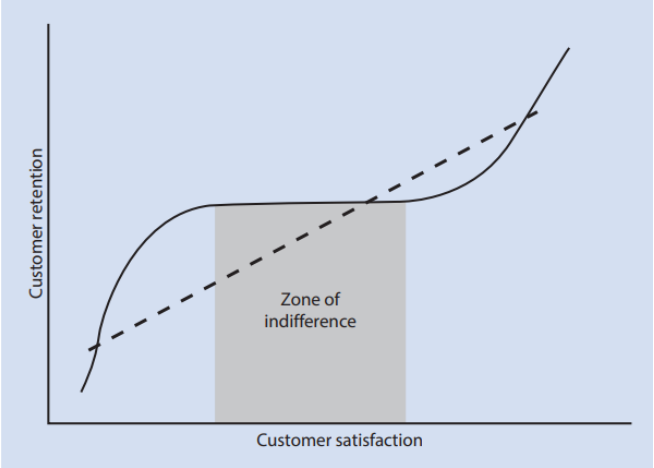


Figura 3: Illustrazione del legame soddisfazione-ritenzione. Nota: la linea tratteggiata rappresenta un'approssimazione lineare della relazione non lineare mostrata. (Anderson & Mittal, 2000)

Il Customer relationship management (CRM) sembra offrire la strategia e la soluzione necessaria per mantenere i clienti felici, sorridenti e legati all'organizzazione per tutta la durata della loro vita. Il CRM ha molteplici sfaccettature e implicazioni per le imprese che ambiscono sempre di più a mantenere e aumentare la propria base clienti, evitare la cannibalizzazione da parte dei concorrenti e mantenere i propri clienti fedeli *(Bhakane, B., 2015).*

Di seguito sono elencate alcune strategie di CRM che sono state adottate a livello globale dalle industrie per un processo di CRM efficace.

a. Migliore gestione dei clienti: Come risulta da varie indagini, alcune aziende non sono in linea con la filosofia del customer care, il che porta a clienti insoddisfatti e a un cattivo passaparola. È necessario prestare molta attenzione affinché ogni organizzazione comprenda appieno l'importanza del servizio al cliente e della sua soddisfazione, e che lo fornisca prontamente ai consumatori. È necessario definire una politica standardizzata di prezzi, sconti e incentivi, in modo che non vi siano disparità nelle offerte ricevute dai clienti nei diversi punti vendita.

b. Riduzione dei tempi di risoluzione dei reclami: le organizzazioni hanno compiuto continui sforzi per ridurre i tempi di risoluzione dei reclami, che sono molto apprezzati dai clienti.

c. Comunicazione regolare con i clienti: L'azienda cerca di mantenere un contatto costante con i propri clienti attraverso telefonate per capire i servizi forniti, le prestazioni, il livello di soddisfazione, ecc. e raccogliere suggerimenti e feedback. Questo crea un senso di appartenenza nella mente dei clienti e l'impressione che l'azienda si preoccupi di loro.

2.3 Il Machine Learning nella churn prevention:

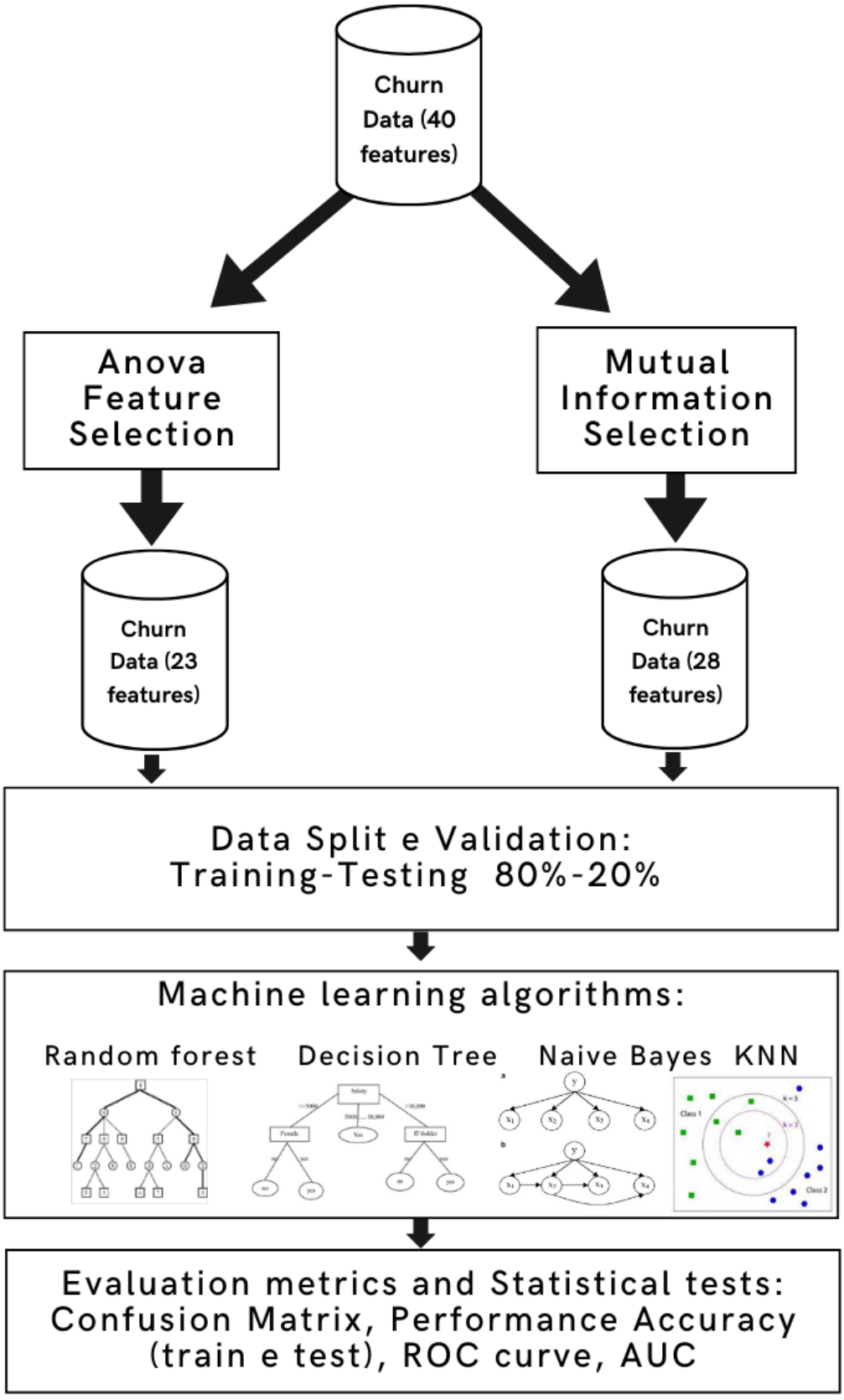
In seguito alla progressiva presa di coscienza dell'importanza della previsione dell'abbandono dei clienti, si è cercato di dare una risposta rigorosa a questo problema. Essa giunge dal campo della data science, in particolare attraverso lo strumento del machine learning: costruire una conoscenza approfondita della natura matematica del problema e conseguentemente individuare modelli predittivi che bene li rappresentino sono i due step che costituiscono la tattica migliore per affrontare preventivamente il fenomeno del churn. Questo è supportato da innumerevoli testimonianze in letteratura: è sufficiente una ricerca anche superficiale per rendersi conto della varietà di tecniche di machine learning che negli anni gli scienziati hanno testato nel tentativo di arginare il problema del customer churn. A questo punto, dunque, resta solo da decidere quale siano le tecniche che meglio si adattino al problema di cui si tratterà in questo lavoro di tesi. Come si potrà vedere meglio nel Capitolo 3, anche attraverso i paper citati in analisi, storicamente i data scientist hanno intrapreso la via del ML in termini di sviluppo di modelli di classificazione per la suddivisione dei clienti nelle due classi dei churners e non churners. A questo risultato si è giunti sperimentando la validità di diversi tipi di modelli. Esistono infatti due approcci diversi all'estrapolazione di informazioni: il supervised learning, che sfrutta la conoscenza della ground thruth - ovvero l'etichetta churner-non churner - per vagliare la validità del modello, e l'unsupervised learning, che al contrario parte dal presupposto dell'ignoranza dell'etichetta di classificazione e cerca di estrapolare regole nascoste che descrivano la somiglianza tra i record. Per quanto riguarda invece il supervised learning, esistono almeno tre categorie di modelli considerabili: modelli semplici di classicazione, modelli ensemble (che reiterano con varie metodologie i modelli semplici al fine di ottimizzarne i risultati) e reti neurali. Le scuole di pensiero in termini di classificazione di questo tipo di dataset si sono divise tra chi favorisce l'uso dei modelli supervised, chi quelli unsupervised e chi ancora ha provato ad unire le due.

3 Caso pratico

Questa sezione fornisce una rassegna esaustiva dello studio svolto che parla di un sistema di predizione del tasso di abbandono degli utenti nel settore energetico. Come anticipato precedentemente, il tasso di abbandono di un'azienda quantifica il numero di clienti che abbandonano un determinato servizio o non acquistano più un determinato prodotto; per le aziende, quindi, è una metrica fondamentale per monitorare il comportamento dei propri clienti e per poter intervenire in modo efficace al fine di massimizzare profitti. In questo studio, in particolare, è stato applicato un approccio di Machine Learning per prevedere il tasso di abbandono dei clienti di un provider energetico della Nuova Zelanda.
L'analisi della previsione dei clienti prossimi all'abbandono permetterà al settore energetico e al dipartimento CRM di identificare quali persone stanno per abbandonare la rete. Per analizzare la seguente previsione in questo lavoro è stato svolto un problema di classificazione binaria. Ciascun abbonato è stato cioè classificato come potenziale churner o non churner. Sono state utilizzate una serie di tecniche di apprendimento automatico per la classificazione di abbandono e non abbandono su tre grandi set di dati forniti da Powerco e, dopo aver fatto le opportune considerazioni delle analisi svolte, è stato individuato il modello più performante in questo task specifico. Tutto il lavoro è stato svolto su Python, uno dei linguaggi di programmazione più comunemente usati poiché costituito da una sintassi semplice e potente che ne facilita l'apprendimento.

Il seguente capitolo è stato strutturato seguendo l'ordine delle fasi svolte durante il lavoro. Nel primo paragrafo è stata riassunta l'analisi dei lavori precedentemente svolti su questo tema. Questa analisi preventiva ha permesso di contestualizzare l'argomento trattato ed analizzare le tecniche che sono state applicate in studi passati. Nel secondo paragrafo è presente la descrizione dei materiali utilizzati ai fini dello studio quali i dati acquisiti e gli strumenti utilizzati per svolgere la sperimentazione. Nel paragrafo successivo sono state descritte le manipolazioni effettuate sui dati e le tecniche statistiche adottate, cioè quell'insieme di operazioni che permettono di preparare e organizzare i dati grezzi prima di avviare l'algoritmo di apprendimento. Successivamente sono state analizzate le tecniche di features selection che sono state utilizzate per eliminare eventuali informazioni ridondanti e migliorare la complessità

computazionale dei modelli. Infine, sono stati descritti i quattro modelli implementati nello studio e le relative metriche per valutare la performance di ciascun algoritmo.


Churn Data (40 features)
Anova Feature Selection
Mutual Information Selection
Churn Data (23 features)
Churn Data (28 features)
Data Split e Validation: Training-Testing 80%-20%
Machine learning algorithms:
Random forest
Decision Tree
Naive Bayes
KNN
Evaluation metrics and Statistical tests: Confusion Matrix, Performance Accuracy (train e test), ROC curve, AUC

3.1 Stato dell'arte:

L'acquisizione e la fidelizzazione dei clienti sono le principali preoccupazioni nel mondo degli affari di oggigiorno. Il rapido aumento del mercato in ogni attività sta portando a un maggior numero di abbonati. Di conseguenza, le aziende attualmente prestano una maggiore attenzione nell'instaurare delle vere e proprie relazioni con i propri clienti. È diventato obbligatorio per i fornitori di servizi ridurre il tasso di abbandono, perché la negligenza potrebbe comportare una riduzione della redditività in una prospettiva sempre più crescente *(Hashmi, N., Butt, N., et al., 2013).* Per questo motivo si è diffusa la convinzione che la migliore strategia di marketing consista nel trattenere gli abbonati già esistenti o più semplicemente ridurre il tasso di abbandono degli stessi *(Golshan Mohammadi et al., 2013*). Per affrontare questo problema, le tecniche di data mining si sono rivelate gli strumenti migliori per combattere il crescente tasso di abbandono dei clienti. Il data mining, come campo principale delle scienze informatiche, è definito come il processo di estrazione di modelli nascosti da insiemi di dati molto grandi utilizzando tecniche statistiche, matematiche, di intelligenza artificiale e tecniche di apprendimento automatico *(Turban, Aronson, Liang, and Sharda, 2007).*

Il data mining è sempre più utilizzato per migliorare l'efficienza del marketing da molte delle più grandi aziende europee e non solo. In senso stretto, il data mining è la scoperta automatica di modelli «interessanti» e non ovvi nascosti in un database che hanno un elevato potenziale di contributo ai profitti. Come viene usata qui, la parola «interessante» ha un significato speciale per la comunità del data mining: le relazioni «interessanti» sono quelle che potrebbero avere un impatto sulla strategia o sulla tattica e, in definitiva, sugli obiettivi di un'organizzazione *(Peacock, P. R., 1998).* Il data mining comprende metodi basati su computer, generalmente chiamati metodi di «apprendimento automatico», che estraggono modelli o informazioni dai dati richiedendo solo un coinvolgimento umano limitato. La maggior parte di questi metodi è di origine relativamente recente e affonda le sue radici nell'intelligenza artificiale (IA).

Sebbene i volumi di dati su cui devono lavorare i responsabili delle decisioni di marketing siano già molto grandi e aumenteranno ancora di più, sono disponibili strumenti che aiutano a risolvere il problema. Ad esempio, l'uso oculato del campionamento probabilistico, della modellazione predittiva e dei metodi di apprendimento automatico può produrre risultati validi riducendo al contempo in modo significativo l'attività di gestione dei dati. Per molte aziende oggi, il data mining sta assumendo un ruolo centrale e trainante per il business. Il processo viene

messo in funzionamento continuo e sta diventando il fulcro delle operazioni aziendali in quanto ha molti utilizzi nel marketing, tra cui l'acquisizione dei clienti, la loro fidelizzazione, l'abbandono dei clienti e l'analisi del portafoglio di mercato.
In questo studio, in particolare, le tecniche di data mining sono state applicate per generare un modello predittivo del tasso di abbandono dei clienti di un provider energetico neozelandese, in modo che le strategie di fidelizzazione e l'azienda possano prosperare massimizzando le entrate complessive.
In studi passati sono già stati applicati metodi di machine learning per la previsione del tasso di abbandono in diversi settori. Il grande vantaggio degli approcci di machine learning rispetto ai metodi tradizionali è che sono metodi all'avanguardia che permettono alle aziende di identificare in modo più accurato i segmenti di clientela più rischiosi in termini di rischio di abbandono e quindi migliorare la loro strategia di fidelizzazione.

Dalla letteratura è emerso che i principali settori in cui sono stati testati modelli predittivi sul tasso di abbandono sono: il settore delle telecomunicazioni (Umman, Tuğba, Şimşek, & Gürsoy,2010; Verbeke et al., 2011; Tarik Rashid ,2008; Adnan Idris et al.,2012; Bingquan Huang et al., 2012; Hung, S. Y.et al., 2006), delle carte di credito (Guangli Nie, Wei Rowe et al., 2011), dei servizi Internet (Afaq Alam Khan et al., 2010; B.Q. Huang et al., 2009; Li, S. T., Shue, L. Y., 2006) , dei servizi bancari (Yaya Xie a, Xiu Li, E.W.T. Ngai b, Weiyun Ying, 2009;Koh, H. C., & Chan, K. L. G., 2002; Au, W. H., & Chan, K. C. C. , 2003; Chiang, D. A., 2002) e dei servizi finanziari (B Larivière, D Van den Poel, 2004).
Ed è stato dimostrato che, tra tutti i settori, le aziende di telecomunicazioni hanno registrato negli ultimi anni il più alto tasso di abbandono annuale, dal 20% al 40% (Hashmi, N., Butt, N. A., et al., 2013).

Uno studio condotto da Ullah, I., Raza, B. et al., propone un modello di previsione del tasso di abbandono che utilizza tecniche di classificazione e di raggruppamento per identificare i clienti in stato di abbandono e fornisce i fattori alla base del tasso di abbandono dei clienti nel settore delle telecomunicazioni. Il modello proposto prima classifica i dati dei clienti churn utilizzando algoritmi di classificazione, in cui l'algoritmo Random Forest (RF) ha funzionato bene con l'88,63% di istanze correttamente classificate. Dopo la classificazione, il modello proposto segmenta i dati del cliente in stato di abbandono categorizzando i clienti in gruppi omogenei per fornire offerte di fidelizzazione basate sul gruppo. Conoscendo i fattori di abbandono più significativi dei dati dei clienti, il CRM è in grado di migliorare la produttività, consigliare

promozioni pertinenti al gruppo di probabili clienti in stato di abbandono sulla base di modelli di comportamento simili e migliorare eccessivamente le campagne di marketing dell'azienda. Il modello di previsione del tasso di abbandono proposto viene valutato utilizzando metriche quali accuratezza, precisione, recall e l'area delle caratteristiche operative di ricezione o anche detta Curva ROC (Ullah, I., Raza, B., Malik, A., et al., 2019).

Analogamente lo studioso Anil Jadhav ha condotto una analisi per prevedere con precisione quali dipendenti hanno intenzione di lasciare l'organizzazione nei prossimi 2 anni, utilizzando alcuni modelli di classificazione come l'SVM, il Naive Bayes e il Random Forest. Dallo studio è emerso che, il classificatore Random Forest è il modello migliore per la previsione del tasso di abbandono dei dipendenti in quanto ha dato la massima accuratezza e valore di recall rispetto agli altri modelli analizzati *(Jadhav, A., 2021).*

Inoltre, l'articolo pubblicato da Agarwal, V., et al., dimostra come la predizione del tasso di interesse abbia campi applicativi anche nel settore bancario. L'articolo, infatti, applica gli algoritmi di apprendimento automatico per identificare i clienti bancari che potrebbero valutare di cambiare istituto finanziario. Nello studio sono stati utilizzati modelli di classificazione, quali la Regressione logistica (LR) e Naïve Bayes (NB) per prevedere con efficacia quali clienti hanno maggiori probabilità di lasciare la banca in futuro utilizzando dati quali l'età, l'ubicazione, il sesso, informazioni sulla carta di credito, il saldo, ecc. Dai risultati della ricerca è emerso che il modello NB fornisce risultati più accurati sul tasso di abbandono rispetto all'algoritmo di Regressione logistica (Agarwal, V., Taware, S., et. al., 2022).

Sebbene gli studiosi nella maggior parte dei lavori passati abbiano utilizzato modelli di apprendimento supervisionato, in alcuni casi hanno sperimentato anche metodi non supervisionati. Ad esempio, nel lavoro svolto dagli studiosi Vijaya, J., e Sivasankar, E., sono stati applicati modelli di classificazione e modelli ibridi. La prima serie di esperimenti è stata condotta per valutare le prestazioni dei modelli di classificazione singoli, come l'albero decisionale (DT), k-nearest neighbor (KNN), linear discriminant analysis (LDA), support vector machine (SVM) e naive bayes (NB). La successiva serie di esperimenti è stata effettuata per valutare le prestazioni dei modelli ibridi gli algoritmi K-Means e K-Medoids insieme agli algoritmi di classificazione decision tree (DT), k-nearest bayes (NB). Successivamente viene fatto un confronto delle tecniche implementate per valutare l'efficacia delle stesse. Negli algoritmi di classificazione, il set di dati proposto viene suddiviso in set di addestramento e set

di test in base al metodo dell'hold-out e le prestazioni calcolate durante la sperimentazione utilizzano metriche quali l'accuratezza e la sensibilità. Dall'analisi delle metriche emerge che il classificatore SVM raggiunge le performance migliori con una sensibilità del 100 per cento e un'accuratezza del 92,55 *(Vijaya, J., & Sivasankar, E., 2018).*

Nel modello ibrido, il set di dati proposto viene segmentato utilizzando il metodo di clustering K-Means; quindi, ciascuno dei cluster viene suddiviso in set di dati di allenamento e di test in base al metodo di hold-out. L'insieme dei dati di addestramento viene modellato utilizzando i classificatori e l'insieme dei dati di test viene predetto in base al modello progettato dai classificatori stessi. I risultati mostrano che tutti i classificatori decision tree (DT), k-nearest neighbor (KNN), linear discriminant analysis (LDA), support vector machine (SVM), naive bayes (NB) insieme al K-Means producono risultati migliori rispetto alle loro prestazioni individuali. Dei cinque modelli ibridi, l'SVM insieme al K-Means produce un'accuratezza, una sensibilità e una specificità migliori.

Dopo aver svolto un'accurata analisi della letteratura di riferimento si può affermare che l'obiettivo di questo studio è quello di contribuire a un reale ampliamento della letteratura disponibile e in particolar modo quello di estendere gli studi sulla previsione del tasso di abbandono nel campo delle multiutility. Un campo in cui gli studi svolti non sono ancora molto estesi rispetto agli altri settori analizzati ma che, proprio come agli altri settori, necessitano delle opportune analisi dei propri clienti. La previsione del tasso di abbandono consentirebbe alle multiutility di elaborare le più opportune strategie di conservazione mirate per limitare le perdite e massimizzare la soddisfazione dei clienti. Ad esempio, tramite questi studi, le aziende potrebbero promuovere specifici incentivi ai segmenti di clientela più rischiosi (ovvero i più inclini a lasciare l'azienda) con l'auspicio che restino fidelizzati.

3.2 Materiali e metodi:

Lo studio in questione analizza il tasso di abbandono dei clienti di un provider energetico per delineare le più opportune strategie di marketing. Il lavoro è stato condotto con il supporto di tecniche di machine learning; tecniche che supportano le aziende nell'identificare, prevedere e fidelizzare i clienti agitati, supportando così il processo decisionale e CRM e massimizzare i propri profitti e rimanere in vita in un mercato competitivo.

L'obiettivo finale di questo studio consiste nell'individuare i clienti in declino che rischiano di cambiare fornitore, in particolare per i clienti del segmento PMI e del mercato europeo, attraverso la questione della liberalizzazione dell'energia.

Verrà prevista la probabilità di abbandono da parte dei clienti e verranno fornite informazioni utili sulla base dei dati disponibili forniti da Powerco.

Come anticipato precedentemente la sperimentazione è stata svolta interamente su Python. Python è un linguaggio di programmazione ideato alla fine degli anni '80 e implementato nel dicembre 1989 da Guido van Rossum presso la CWI nei Paesi Bassi come successore del linguaggio ABC in grado di gestire le eccezioni e di interfacciarsi con il sistema operativo Amoeba. *(Hao, J., Ho, T. K., 2019).*

Il linguaggio di programmazione Python attualmente sta guadagnando un'enorme popolarità tra i *data scientist* e gli sviluppatori di software. A differenza del linguaggio di programmazione R, destinato principalmente all'analisi statistica dei dati, Python si presenta in una gamma molto più ampia di applicazioni, come lo sviluppo di siti Internet o applicazioni Web e desktop, l'accesso a database, il calcolo scientifico e lo sviluppo di software e giochi.

Esistono principalmente due versioni di Python: le versioni 2.x e 3.x. La versione 2.x è una versione considerata ormai obsoleta, il cui supporto e la manutenzione sono stati interrotti intorno al 2020. Al contrario, la versione 3.x è una riprogettazione basata sulla versione 2.x ed è considerata essere il futuro di Python.

Python non è un linguaggio compattato, il che significa che non precompila il codice in formato binario. Un ambiente software, l'interprete di Python, traduce lo script in binario durante l'esecuzione del codice in tempo reale. Il linguaggio di programmazione viene fornito con alcune funzionalità di base, ma si affida a pacchetti esterni per eseguire quasi tutti i calcoli

numerici. Dopo la naturale selezione avvenuta negli ultimi 10 anni, un piccolo insieme di pacchetti che forniscono alcune

capacità di calcolo fondamentali ha ricevuto un'ampia accettazione nella comunità Python *(Hao, J., Ho, T. K., 2019).* Nella tabella sottotsante sono elencati alcuni dei pacchetti fondamentali che svolgono il lavoro più pesante e sono mantenuti dalla comunità Python. Oltre a questi pacchetti fondamentali, il front-end dell'interprete Python è in continua evoluzione.

NOME DEL PACCHETTO	**DESCRIZIONE**	**SITO**
NUMPY	Pacchetto fondamentale per il calcolo numerico	http://www.numpy.org/
SCIPY	Pacchetto avanzato per il calcolo scientifico basato su NumPy	https://www.scipy.org/
PANDAS	Pacchetto avanzato per la struttura e la manipolazione dei dati	http://pandas.pydata.org/
MATPLOTLIB	Il pacchetto fornisce funzionalità di plottaggio di base	https://matplotlib.org/
SCIKIT-LEARN	Pacchetto per l'apprendimento automatico, spesso abbreviato in sklearn	http://scikit-learn.org/

Proprio come in questo caso, il modo più semplice per ottenere Python, i pacchetti principali e Jupyter Notebook è quello di installarli attraverso la suite anaconda (https://www.anaconda.com/download),

che raccoglie un insieme selezionato di pacchetti Python e fornisce programmi di installazione per

Windows, MacOS e Linux. Dopo aver scaricato il programma di installazione di anaconda e installare le suite Python, si devono seguire le istruzioni sul sito web di anaconda per lanciare il front-end interattivo, Jupyter Notebook. A questo punto, il linguaggio è pronto per la sperimentazione.

Lo studio che è stato svolto può essere sintetizzato in cinque fasi ben distinte:

- Acquisizione dei dati: Questa fase consiste nell'acquisizione dei dati necessari per condurre l'analisi. Il set di dati per questo studio è stato acquisito da Powerco ed è stato utilizzato per analizzare le tendenze di marketing dei clienti della società neozelandese.
- Esplorazione del dataset: Lo scopo principale dell'analisi esplorativa dei dati è quello di aiutare a guardare ai dati prima di fare qualsiasi supposizione. Può aiutare a identificare gli errori e a comprendere meglio i modelli all'interno dei dati, rilevare i valori e gli eventi anomali e a trovare interessanti relazioni tra le variabili.
- Data Pre-processing: La preelaborazione dei dati è la fase più importante nei modelli di previsione, poiché i dati sono costituiti da ambiguità, errori e ridondanze che devono essere eliminate in anticipo. I dati raccolti vengono prima aggregati e poi puliti, poiché i dati raccolti non sono adatti ai fini della modellazione.
- Estrazione delle features: In questa fase vengono identificati gli attributi per il processo di classificazione. In questo lavoro, sono state utilizzate sia variabili numeriche sia categoriche.
- Scelta del modello e metriche: In questa fase finale vengono confrontati i diversi modelli di predizione e, con il supporto delle metriche, si definisce il modello di classificazione più affidabile.

3.2.1 Acquisizione dei dati:

Nel seguente studio è stato utilizzato un set di dati fornito da *Powerco*, uno dei clienti di BCG che si occupa della fornitura di gas ed elettricità per le PMI (Piccole Medie Imprese) e per i clienti residenziali in Nuova Zelanda. L'obiettivo finale di questo studio consiste nell'individuare i clienti in declino che rischiano di cambiare fornitore, in particolare per i clienti del segmento PMI e del mercato europeo, attraverso la questione della liberalizzazione dell'energia.

Powerco è nata in seguito alle riforme energetiche degli anni '90 in Nuova Zelanda. La sua storia risale a partire da una serie di enti locali per l'energia elettrica e società del gas che operavano in tutta l'Isola del Nord. L'azienda è attualmente di proprietà di società australiane, tra cui Queensland Investment Corporation e AMP Limited. Powerco è il secondo distributore di gas e di elettricità della Nuova Zelanda. È una delle due sole società a distribuire sia elettricità che gas naturale attraverso la propria rete. La sua rete fornisce elettricità e gas alle famiglie di tutta l'Isola del Nord dalla rete nazionale di trasmissione dell'elettricità Transpower e dal sistema di trasmissione del gas naturale di proprietà e gestito da Vector Limited.
Powerco è un distributore la cui attività è completamente separata dalla generazione, dalla trasmissione nazionale e dalla vendita al dettaglio (vendita di elettricità consegnata agli utenti finali). Gestisce una rete di distribuzione elettrica locale di 30.000 km che rifornisce 320.000 famiglie, industrie e imprese dalla rete nazionale di Transpower e una rete di distribuzione di gas naturale di 6.000 km che rifornisce 102.000 famiglie, industrie e imprese dalla rete di trasmissione di Vector.

Inizialmente i dati forniti da Powerco erano suddivisi in tre dataset distinti che contenevano rispettivamente:

1. Dati storici dei clienti: dati dei clienti, come l'utilizzo, la data di sottoscrizione, le previsioni di utilizzo.
2. Dati storici sui prezzi: dati storici sui prezzi, variabili e fissi.
3. Indicatore di abbandono: dati che indicano se ogni cliente ha effettuato il churning o meno.

I tre dataset sono poi stati uniti a partire dall' ID cliente, che è la variabile in comune tra tutti e tre i dataset.

Descrizione delle variabili:

- id: contatto id
- activity_new category: categoria dell'attività svolta dalla compagnia
- campaign_disc_ele: codice della campagna elettrica a cui il cliente ha aderito per ultima
- channel_sales: codice del canale di vendita
- cons_12m electricity: consumo di elettricità negli ultimi 12 mesi
- cons_gas_12m: consumo di gas negli ultimi 12 mesi
- cons_last_month: consumo di elettricità nell'ultimo mese
- date_activ: data di attivazione del contratto
- date_end: data di scadenza del contratto
- date_first_activ: data di attivazione del primo contratto del cliente
- date_modif_prod: data dell'ultima modifica del prodotto
- date_renewal: data del prossimo rinnovo contrattuale
- forecast_base_bill_ele: previsione della bolletta elettrica di base per il mese successivo
- forecast_base_bill_year: previsione della bolletta elettrica di base per l'anno solare
- forecast_bill_12m: previsione della bolletta elettrica di base per 12 mesi
- forecast_cons: consumo di elettricità previsto per il mese successivo
- forecast_cons_12m: consumo di elettricità previsto per i prossimi 12 mesi
- forecast_cons_year: consumo di elettricità previsto per il prossimo anno solare
- forecast_discount_energy: valore previsto di sconto corrente
- forecast_meter_rent_12m: bolletta prevista per il noleggio del contatore per i prossimi 12 mesi
- forecast_price_energy_p: prezzo dell'energia previsto per il primo periodo
- forecast_price_energy_p2: prezzo dell'energia previsto per il secondo periodo
- forecast_price_pow_p1: Prezzo dell'energia previsto per il primo periodo
- has_gas: viene indicato se il cliente è anche cliente gas
- imp_cons: consumo corrente pagato

- margin_gross_pow_ele: margine lordo sull'abbonamento di energia elettrica
- margin_net_pow_ele: margine netto sull'abbonamento di energia elettrica
- nb_prod_act: Numero di prodotti e servizi attivi
- net_margin: Margine netto totale
- num_years_antig: Anctichità del cliente (in numero di anni)
- origin_up: Codice della campagna elettrica a cui il cliente ha aderito per la prima volta
- pow_max: Potenza sottoscritta
- price_date: Data di riferimento
- price_p1_var: Prezzo dell'energia per il primo periodo
- price_p2_var: Prezzo dell'energia per il secondo periodo
- price_p3_var: Prezzo dell'energia per il terzo periodo
- price_p1_fix: Prezzo dell'energia elettrica per il primo periodo
- price_p2_fix: Prezzo dell'energia elettrica per il secondo periodo
- price_p3_fix: Prezzo dell'energia elettrica per il terzo periodo
- churned: Il cliente ha effettuato il churning nei tre mesi successivi

Nel corso della sperimentazione è stata individuata come caratteristica target la variabile "churned". Le istanze al termine dello studio saranno quindi classificate come *churned* (1) o non *churned* (0).

Una volta acquisiti i dati, prima ancora di passare alla fase di esplorazione del dataset, sono state importate le librerie su Python che risultano necessarie per importare, lavorare ed analizzare i dati. In particolare, in questo studio sono state importate le seguenti librerie:

```
#Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

- *NumPy* è una libreria di funzioni scientifiche open source per il linguaggio di programmazione Python, che fornisce supporto a grandi matrici e array multidimensionali insieme a una vasta collezione di funzioni matematiche di alto livello per poter operare efficientemente su queste strutture dati.

- *Pandas* è una libreria del linguaggio Python utilizzata per l'analisi e la manipolazione dei dati. In particolare, offre strutture dati e operazioni per manipolare tabelle numeriche e serie temporali. Questa libreria verrà utilizzata durante tutto il corso dell'analisi.

- La libreria *Matplotlib* verrà utilizzata per la creazione di grafici nel corso dello studio, come ad esempio grafici a barre e boxplot.

- *Seaborn* è una libreria di Python che potenzia gli strumenti di data visualization del modulo matplotlib. Nel modulo Seaborn sono presenti diverse funzionalità per rappresentare graficamente i dati. In particolar modo sono presenti dei metodi che agevolano la costruzione dei grafici statistici con matplotlib.

  Infine, è stato importato il pacchetto *SciKit-learn* ed i suoi relativi moduli.

- *SciKit-learn* è una libreria di apprendimento automatico che è stata implementata per l'utilizzo degli algoritmi di classificazione e le relative metriche. In generale il pacchetto contiene algoritmi di classificazione, regressione e clustering e macchine a vettori di supporto, regressione logistica, classificatore bayesiano ed è progettato per operare con le librerie Numpy e Scipy. In questo studio il modulo sarà particolarmente utile per utilizzare gli algoritmi Random Forest, Decision Tree, Naive Bayes, KNN e le relative metriche, oltre che per applicare le opportune tecniche di estrazione delle caratteristiche. In generale, Scikit-learn è il pacchetto per l'apprendimento automatico più completo e open source in Python.

```
#Sklearn modules
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import classification_report, confusion_matrix,
plot_confusion_matrix, roc_curve, roc_auc_score
from sklearn.model_selection import train_test_split, cross_val_score,
StratifiedKFold, GridSearchCV
```

```
from sklearn.feature_selection import SelectKBest, f_classif, chi2,
mutual_info_regression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.decomposition import PCA
from sklearn.naive_bayes import GaussianNB
```

3.2.2. Esplorazione del dataset:

L'analisi esplorativa dei dati (o EDA, exploratory data analysis) permette di effettuare uno studio preventivo dei dati ottenuti e su cui si intende eseguire la sperimentazione. L'analisi esplorativa, quindi, può essere utilizzata per assicurarsi che i risultati prodotti siano validi e applicabili a qualsiasi risultato e obiettivo di business desiderato.
Il processo di esplorazione del dataset può essere sintetizzato principalmente in due fasi:

- Screening dei dati,
- Identificazione dei valori anomali.

In primo luogo, è stato eseguito lo *screening* dei dati che permette di visualizzare i dati e comprenderne la natura. Sulla base di questa analisi possiamo affermare che il dataset è composto da 193002 righe e 40 colonne al cui interno sono presenti complessivamente 40 variabili di natura diversa. È stato possibile poi identificare il *type* di ciascuna variabile; delle 40 variabili si identificano rispettivamente 22 variabili di tipo *float64* e cioè numeri decimali che occupano 64 bit nella memoria del computer, 11 di tipo *object* che fanno riferimento a variabili categoriche e le restanti 7 di tipo *int64* e cioè numeri interi. In questa fase è particolarmente rilevante individuare le variabili di tipo "object" poiché nelle fasi successive saranno oggetto di trasformazioni e modifiche per far sì che vengano date in pasto correttamente all'algoritmo.

Successivamente sono stati individuati i valori mancanti all'interno del dataset che ammontano complessivamente a 1130954. I valori mancanti (missing values) sono valori che, per svariate ragioni, non sono presenti in un dataset. In Python, i valori mancanti sono rappresentati come NaN (Not a Number), None o NULL. Nella fase di esplorazione del dataset è fondamentale individuare questi valori per poi modificarli o eliminarli nella fase successiva in modo da non intaccare l'affidabilità delle previsioni.

Inoltre, con questa analisi è possibile affermare che complessivamente le variabili che hanno il maggior numero di valori mancanti sono: "campaign_disc_ele" (193002), "date_first_activ" (150960), "forecast_base_bill_ele" (150960), "forecast_base_bill_year" (150960), "forecast_bill_12m" (150960), "forecast_cons" (150960), "activity_new" (114432) che detengono più del 50% di valori nulli/ mancanti. Queste variabili sono poi seguite da "channel_sales" (50595), "date_modif_prod" (1875), "forecast_discount_energy" (1507), "forecast_price_energy_p1" (1507), "forecast_price_energy_p2" (1507),

forecast_price_pow_p1 (1507), “price_p1_var” (1359), “price_p2_var” (1359), “price_p3_var” (1359), “price_p1_fix” (1359), “price_p2_fix” (1359), “price_p3_fix” (1359), “origin_up” (1042), “date_renewal” (477), “net_margin” (180), “margin_gross_pow_ele” (156), margin_net_pow_ele (156), “pow_max” (36).

Successivamente, oltre ai valori mancanti, sono stati individuati i valori anomali (o outliers) per ciascuna variabile. Gli outliers, in un insieme di osservazioni, sono valori anomali e aberranti, ossia valori chiaramente distanti dalle altre osservazioni disponibili (Zimek, A., Filzmoser, 2018).
Esistono diverse tecniche per esaminare la presenza di outlier:

- Tecniche di visualizzazione dei dati per esaminare la distribuzione dei dati e verificare la presenza di outlier.
- Metodi statistici per calcolare i punti di dati anomali.
- Metodi statistici per eliminare o trasformare gli outlier.

In questo studio è stato adottato un processo che combina più tecniche, combinando tecniche di visualizzazione e metodi statistici.
In primo luogo sono state analizzate le statistiche descrittive con il comando *.describe()* che ci ha permesso di generare alcune statistiche di riepilogo. L’analisi delle statistiche permette di determinare facilmente se il set di dati presenta o meno dei valori outlier. E in particolare, se dal confronto tra la media e il valore massimo di una variabile emerge un forte scostamento dei valori si può affermare quasi certamente che siano presenti degli outlier poiché il valore della media può essere influenzato dagli outlier.

Index	ast meter rent	cast price energy	cast price energy	ecast price pow	imp cons	ain gross pow	ain net pow	nb prod act	net margin
count	193002	191495	191495	191495	193002	192846	192846	193002	192822
mean	70.2978	0.135906	0.0529447	43.5333	196.15	22.464	21.4634	1.3478	217.965
std	79.0132	0.0262528	0.0486171	5.21229	494.497	23.7013	27.9191	1.46023	366.816
min	-242.96	0	0	-0.122184	-9038.21	-525.54	-615.66	1	-4148.99
25%	16.23	0.115237	0	40.6067	0	11.97	11.95	1	51.97
50%	19.435	0.142881	0.086163	44.3114	44.51	21.09	20.97	1	119.67
75%	131.47	0.146348	0.098837	44.3114	218.07	29.64	29.64	1	275.75
max	2411.69	0.273963	0.195975	59.4447	15042.8	374.64	374.64	32	24570.7

Figura 4: statistiche descrittive di ciascuna variabile del dataset

Come si può notare, ad esempio, la colonna “forecast_meter_rent” presenta valori anomali. Il valore massimo è pari a 2411.69 mentre la media è 70.297. Il fatto che il valore medio sia così piccolo rispetto al valore massimo indica che si tratta di un outlier poiché la media e un valore che tende ad essere influenzato dalla presenza degli outlier.

Successivamente, dopo aver analizzato le statistiche descrittive, è stato applicato uno dei principali strumenti di visualizzazione per individuare gli outlier nei dati: i box plot.
I box plot sono utili perché mostrano i valori minimi e massimi, la mediana e l'intervallo interquartile dei dati. Nel grafico, gli outlier sono rappresentati come punti, il che li rende facilmente visibili.

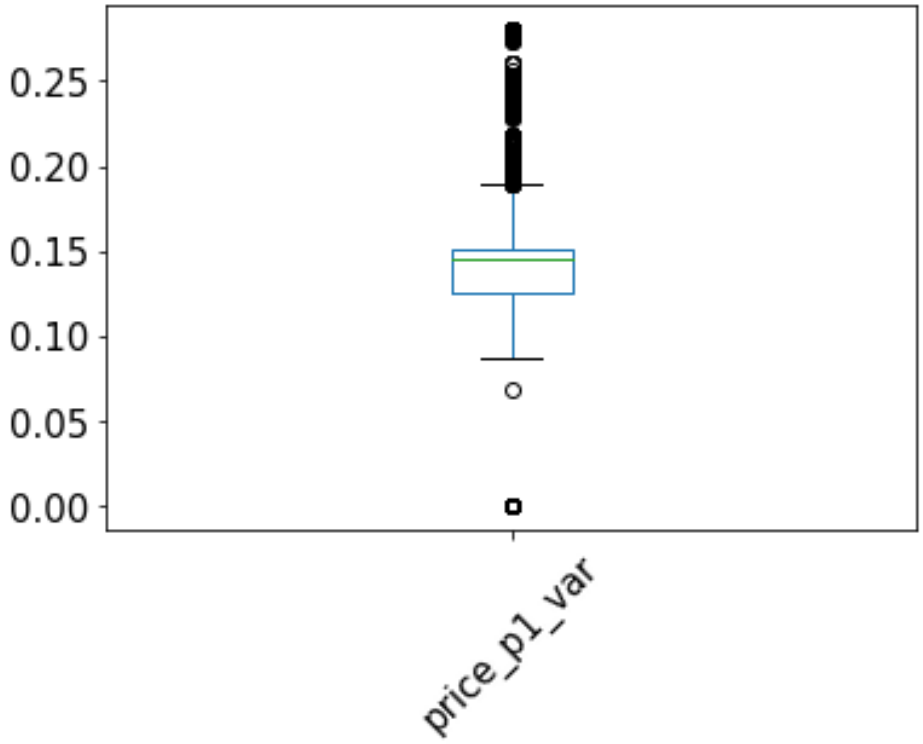


Figura 5: boxplot variabile "price_p1_var" per l'analisi degli outliers

Nel grafico sopra riportato, ad esempio, è facilmente intuibile che i valori superiori a 0.19 si comportino come valori anomali. Osservando semplicemente il box plot possiamo quindi affermare che la variabile "price_p1_var" presenta dei valori anomali. Questo approccio è stato adottato per analizzare la presenza di eventuali outlier in ciascuna variabile e nel complesso è emerso che le variabili del dataset che presentano outlier sono tredici e sono rispettivamente: "cons_12m", "cons_gas_12m", "cons_last_month", "forecast_cons_12m", "forecast_cons_year", "forecast_meter_rent", "forecast_price_energy_p1", "imp_cons", "margin_gross_pow_ele", "margin_net_pow_ele", "net_margin", "pow_max", "price_p1_var".

Una volta analizzate le variabili all'interno del dataset e individuate le anomalie da correggere è stata condotta la fase di preelaborazione dei dati.

Variabile	Descrizione	Tipo variabile	Numero NanOutliers	Outliers	Valori negativi
id	contatto id	object			
activity_new	categoria dell'attività della compagnia	object	114432		
campaign_disc_ele	codice della campagna elettrica a cui il cliente ha aderito per ultimo	float64	193002		
channel_sales	codice del canale di vendita	object	50595		
cons_12m	consumo di elettricità degli ultimi 12 mesi	int64		/	
cons_gas_12m	consumo di gas degli ultimi 12 mesi	int64		/	
cons_last_month	consumo di elettricità dell'ultimo mese	int64		/	/
date_activ	data di attivazione del contratto	object			
date_end	data registrata di scadenza del contratto	object	21		
date_first_activ	data del primo contratto del cliente	object	150960		
date_modif_prod	data dell'ultima modifica del prodotto	object	1875		
date_renewal	data del prossimo rinnovo contrattuale	object	477		
forecast_base_bill_ele	previsione della bolletta elettrica di base per il mese successivo	float64	150960		
forecast_base_bill_year	previsione della bolletta elettrica di base per l'anno solare	float64	150960		
forecast_bill_12m	previsione della bolletta elettrica di base per 12 mesi	float64	150960		
forecast_cons	consumo di elettricità previsto per il mese successivo	float64	150960		
forecast_cons_12m	consumo di elettricità previsto per i prossimi 12 mesi	float64		/	
forecast_cons_year	consumo di elettricità previsto per il prossimo anno solare	int64		/	
forecast_discount_energy	valore previsto di sconto corrente	float64	1507		
forecast_meter_rent_12m	bolletta prevista per il noleggio del contatore per i prossimi 12 mesi	float64		/	
forecast_price_energy_p1	prezzo dell'energia previsto per il 1° periodo	float64	1507	/	
forecast_price_energy_p2	prezzo dell'energia previsto per il 2° periodo	float64	1507		
forecast_price_pow_p1	prezzo previsto dell'energia elettrica per il 1° periodo	float64	1507		/
has_gas	indicato se il cliente è anche cliente gas	object			
imp_cons	consumo corrente pagato	float64		/	
margin_gross_pow_ele	margine lordo sull'abbonamento energia elettrica	float64	156	/	/
margin_net_pow_ele	margine netto sull'abbonamento energia elettrica	float64	156	/	/
nb_prod_act	numero di prodotti e servizi attivi	int64			
net_margin	margine netto totale	float64	180	/	
num_years_antig	antichità del cliente (in numero di anni)	int64			
origin_up	codice campagna elettrica a cui il cliente ha aderito per la prima volta	object	1042		
pow_max	potenza sottoscritta	float64	36	/	
price_date	data di riferimento	object			
price_p1_var	prezzo dell'energia per il 1° periodo	float64	1359	/	
price_p2_var	prezzo dell'energia per il 2° periodo	float64	1359		
price_p3_var	prezzo dell'energia per il 3° periodo	float64	1359		
price_p1_fix	prezzo dell'energia elettrica per il 1° periodo	float64	1359		/

price_p2_fix	prezzo dell'energia elettrica per il 2° periodo	float64	1359
price_p3_fix	prezzo dell'energia elettrica per il 3° periodo	float64	1359
churned	il cliente ha effettuato il churning nei 3 mesi successivi	int64	

3.2.3. Data PreProcessing:

La fase di preelaborazione si riferisce all'insieme di trasformazioni applicate ai dati prima di darli in pasto all'algoritmo. Le tecniche di Data PreProcessing riguardano principalmente la trasformazione e la pulizia dei dati di input e rappresentano uno step necessario e fondamentale per ottenere l'accuratezza richiesta nell'output finale.

La pulizia dei dati è il processo di rilevamento e correzione (o rimozione) di valori corrotti o imprecisi da un set di dati. Più precisamente si riferisce all'identificazione di parti incomplete, errate, imprecise o irrilevanti dei dati e quindi alla sostituzione, modifica o l'eliminazione dei dati sporchi o grossolani.
In questo caso specifico sono stati eliminati i dati non rilevanti ai fini dello studio quali, ad esempio, l'id del cliente e la data di rinnovo del contratto in quanto non concorrono a determinare il tasso di abbandono dei clienti del provider energetico.
Oltre ai dati non necessari, sono state eliminate le variabili che detengono più del 50% di valori nulli che, come possiamo vedere dal grafico a barre, sono: "activity_new", "campaign_disc_ele", "date_first_activ", "forecast_base_bill_ele", "forecast_base_bill_year", "forecast_bill_12m," forecast_cons".

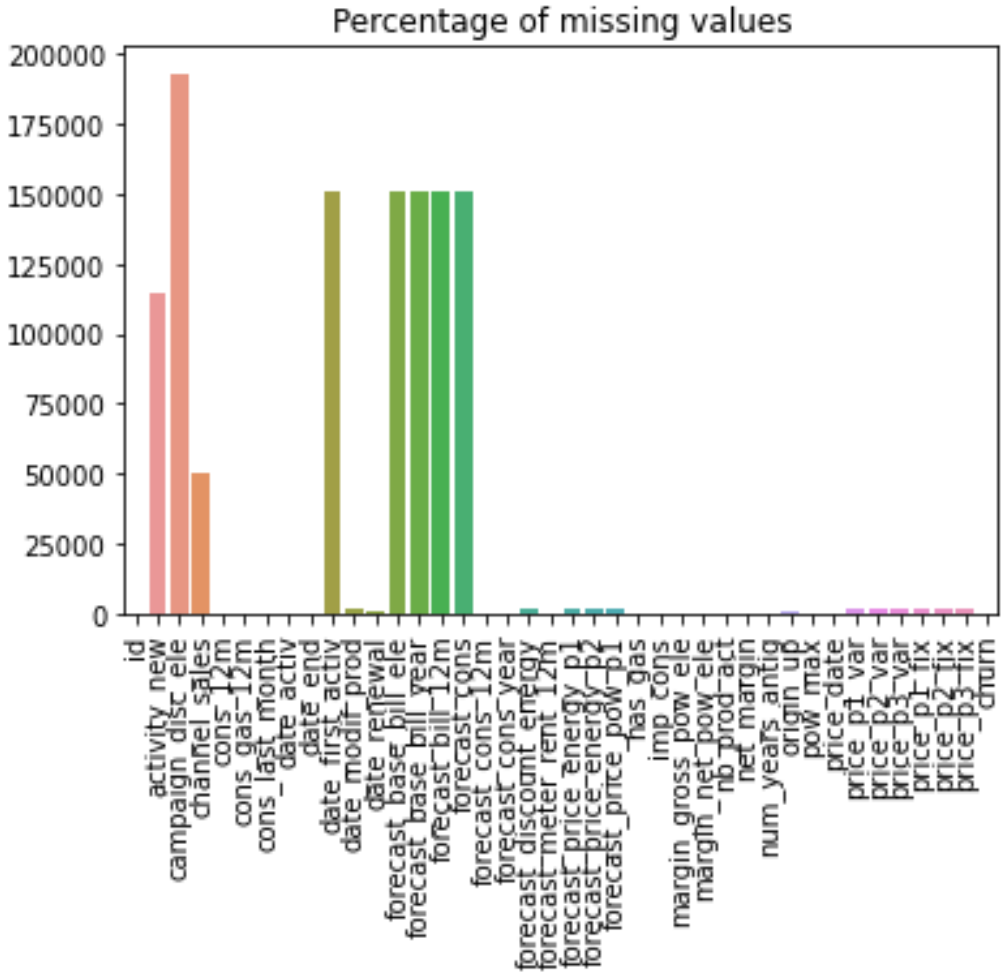


Figura 6: grafico a barre dei valori mancanti per ciascuna variabile, espressi in percentuale

Indipendentemente dalla loro origine, i valori mancanti possono introdurre pregiudizi nell'analisi, ridurre la potenza dei modelli di apprendimento automatico e in generale influire sull' accuratezza delle previsioni. Per questi motivi, la gestione dei valori mancanti è un passaggio fondamentale in questa fase.

Al contrario, le variabili che presentano un numero di valori mancanti inferiore al 50% non sono state eliminate. I valori mancanti, in questo caso sono stati trattati diversamente; essi infatti sono stati sostituiti con la moda o con la mediana a seconda che si trattassero di variabili categoriche o numeriche.

Analogamente, gli *outlier* possono avere un forte impatto sull'analisi statistica e sull'apprendimento automatico, in quanto incidono su calcoli come la media e la deviazione standard e possono influenzare i test di ipotesi. Proprio per questo motivo in questa fase di

preelaborazione i valori anomali, una volta individuati tramite l'utilizzo dei box plot, sono stati sostituiti con i valori di moda e mediana, proprio come nel caso dei *missing values*.

Successivamente è stata effettuata la codifica delle variabili categoriche. Per far sì che un modello operi correttamente è necessario che i dati di natura qualitativa vengano convertiti in forma numerica; per questo motivo, prima di mandare i dati in pasto al modello, è necessario applicare le opportune tecniche di *encoding*.
Con il comando "features_categoriche = df4.select_dtypes(include="object")" sono state individuate e selezionate esclusivamente le variabili categoriche che complessivamente sono cinque: "channel_sales", "date_modif_prod", "date_renewal", "has_gas", "origin_up".

Esistono diverse tecniche per convertire i valori categoriali in valori numerici. In questo studio sono state applicate principalmente due tecniche denominate rispettivamente *Label-Encoder* e *One-Hot Encoding*.

Per le variabili "channel_sales", "date_modif_prod", "date_renewal" e "origin_up" è stata applicata la tecnica Label-Encoder presente all'interno del pacchetto di skit-learn. In questa tecnica il valore categorico viene sostituito con un valore numerico compreso tra 0 e il numero di classi meno 1. Se, ad esempio, il valore della variabile categorica contiene 5 classi distinte, esse vengono numerate da 0 a 1: (0, 1, 2, 3 e 4).
Per la variabile "has_gas" invece è stata utilizzata la tecnica One Hot Encoding. In questo approccio, per ogni categoria di una caratteristica, si crea una nuova colonna chiamata variabile dummy con codifica binaria (0 o 1) per indicare se una particolare riga appartiene a questa categoria. In questo caso specifico con il valore 1 si indicano i clienti che usufruiscono anche del servizio di erogazione del gas oltre che quello energetico; mentre con il valore 0 i clienti che non usufruiscono anche del servizio aggiuntivo.
Con il comando df4.info(), infine, è stata verificata la corretta codifica di tutte le variabili categoriche e che non vi fossero ulteriori variabili da convertire.

Una volta terminato il processo di encoding è stata estratta la y che nel nostro caso è la variabile "churn". Estraendo la variabile di interesse è possibile fare le opportune considerazioni e prevedere il tasso di abbandono dei clienti del provider.
Una volta estratta la variabile dal dataset, la stessa è stata rimossa dal dataset di appartenenza con il comando .drop('churn', axis=1*)*. A questo punto il dataset è composto da tutte le variabili

ripulite e trasformate pronte per essere mandate in pasto al modello meno la variabile churn su cui si intende fare le opportune previsioni.

Una volta analizzato il dataset ed eseguite le opportune modifiche sulle variabili, sono state condotte le analisi statistiche. In questa fase le analisi statistiche permettono di ridurre il volume dei dati pur mantenendo la qualità dell'analisi.
Come affermato da alcuni studiosi, infatti, la riduzione della dimensionalità (o anche detta *Dimensionality Reduction*) si riferisce semplicemente al processo di riduzione del numero di attributi in un set di dati, mantenendo il più possibile la variazione del set di dati originale *(Reddy, G. T., Reddy, M., et al., 2020).*
La riduzione della dimensionalità identifica e rimuove le funzionalità che danneggiano le prestazioni del modello di apprendimento automatico o che non contribuiscono alla sua precisione. Esistono diverse tecniche di dimensionalità, ognuna delle quali è utile per determinate situazioni.

In questo studio, in primo luogo, è stata condotta l'analisi di correlazione per individuare le caratteristiche che sono altamente correlate con la variabile target (churn) e che contribuiscono maggiormente alla varianza del set di dati. L'analisi di correlazione per la riduzione della dimensionalità è una tecnica di riduzione della dimensionalità che consente di identificare correlazioni e modelli in un set di dati in modo che possa essere trasformato in un set di dati di dimensioni notevolmente inferiori senza perdita di informazioni importanti.

Più in generale, l'analisi di correlazione è un'analisi statistica che permette di indagare la relazione lineare esistente tra due variabili quantitative.
In questo studio, infatti, l'analisi di correlazione ci ha permesso di individuare la relazione tra ciascuna variabile del dataset. Questa analisi è importante perché se due variabili sono correlate ciò significa che contribuiscono a fornire le medesime informazioni rendendo l'analisi ridondante; per questo motivo una delle due variabili correlate può essere eliminata per evitare la ridondanza del dataset. I valori all'interno della matrice di correlazione vanno da 1 a -1. Ciò significa che la correlazione può essere positiva o negativa. Nel primo caso, all'aumentare dei valori di una variabile, aumenteranno anche quelli dell'altra. Viceversa, nel caso di correlazioni negative, all'aumentare dei valori della prima variabile, quelli della seconda diminuiranno. Per

conoscere la direzione della correlazione è necessario osservare il segno dell'indice di correlazione.
Dai risultati ottenuti dall'analisi è possibile affermare che le variabili altamente correlate (con soglia > 0.9) sono: cons_last_month e cons_12m, forecast_cons_year e imp_cons, price_p2_var e forecast_price_energy_p2, price_p1_fix e forecast_price_pow_p1, price_p2_fix e price_p3_var, price_p3_fix e price_p3_var, price_p3_fix e price_p2_fix.

Per facilitare la visibilità della correlazione tra le variabili e tra queste ultime e la variabile target è stata utilizzata la mappa di calore.

Una mappa di calore è una rappresentazione bidimensionale dei dati in cui i valori sono rappresentati da colori. La mappa di calore della correlazione è un grafico bidimensionale della quantità di correlazione (misura della dipendenza) tra le variabili rappresentate dai colori. L'intensità variabile del colore rappresenta la misura della correlazione.
Inoltre, la mappa di calore illustra la covarianza tra le diverse caratteristiche ed è una buona guida per trovare ed eliminare le caratteristiche eccessive.
Lo stesso strumento può aiutare a visualizzare le correlazioni tra le caratteristiche e la variabile di destinazione. Questo aiuta a rimuovere le variabili che non influenzano l'obiettivo.

In questo caso la mappa di calore ha lo scopo di mostrare se esiste una relazione tra le variabili come il canale di vendita, il consumo medio di energia per cliente il tutto rispetto al tasso di abbandono del servizio da parte degli stessi clienti. La mappa di calore è stata tracciata tramite l'utilizzo del pacchetto *Seaborn* di Python e per determinare la correlazione è stato utilizzato il metodo corr() di *Pandas*. sns.heatmap(df.corr()). Come possiamo vedere, la mappa mostra colori diversi a seconda del tipo di correlazione: i colori più chiari esprimono una correlazione positiva mentre i colori più scuri indicano una correlazione negativa.

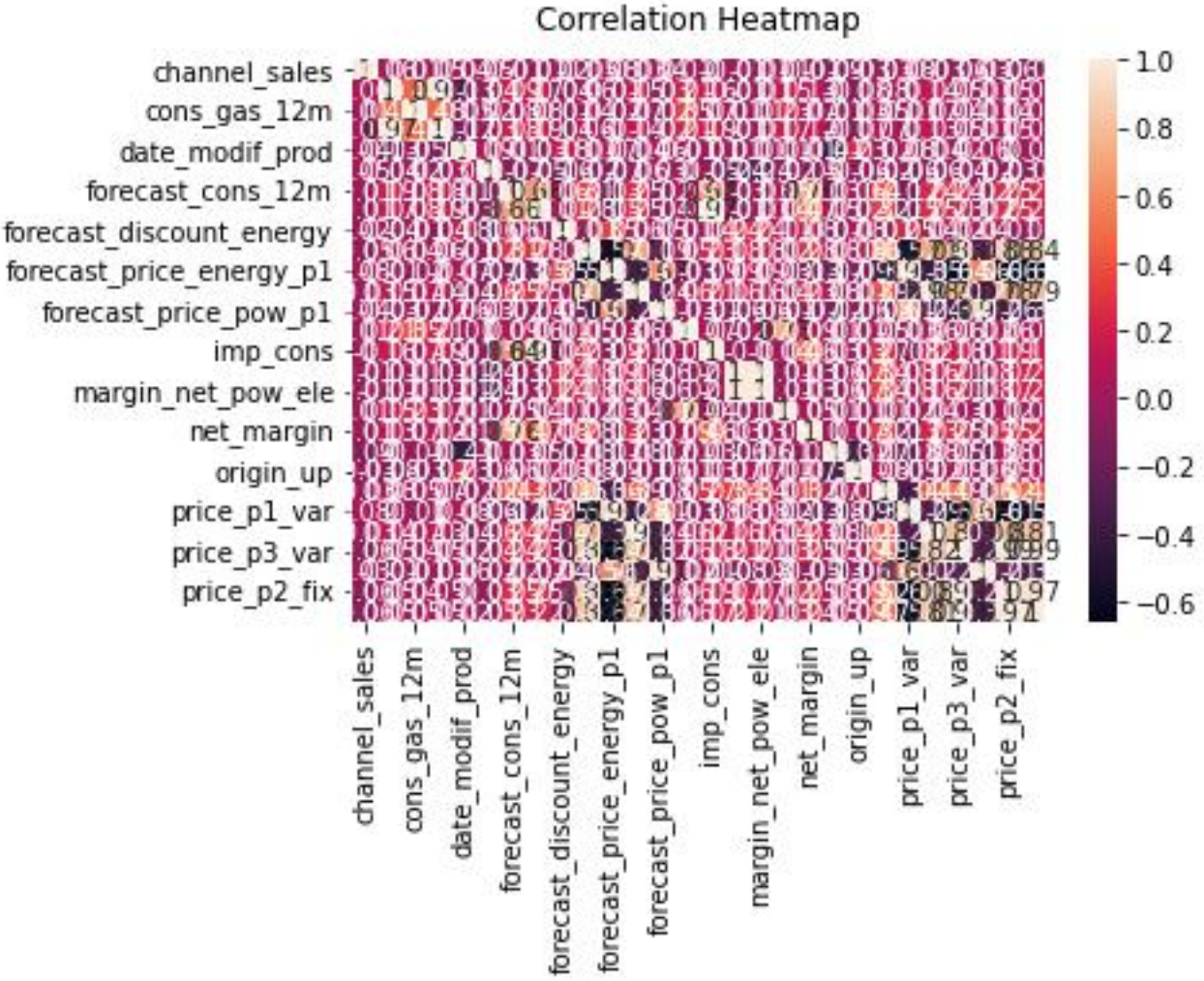


Figura 7: mappa di calore di correlazione

In un secondo momento sono state eliminate le caratteristiche meno correlate rispetto alla variabile obbiettivo e quindi al tasso di abbandono poichè non forniscono informazioni utili per la predizione in analisi ed è stata ottenuta la mappa di calore sottostante:

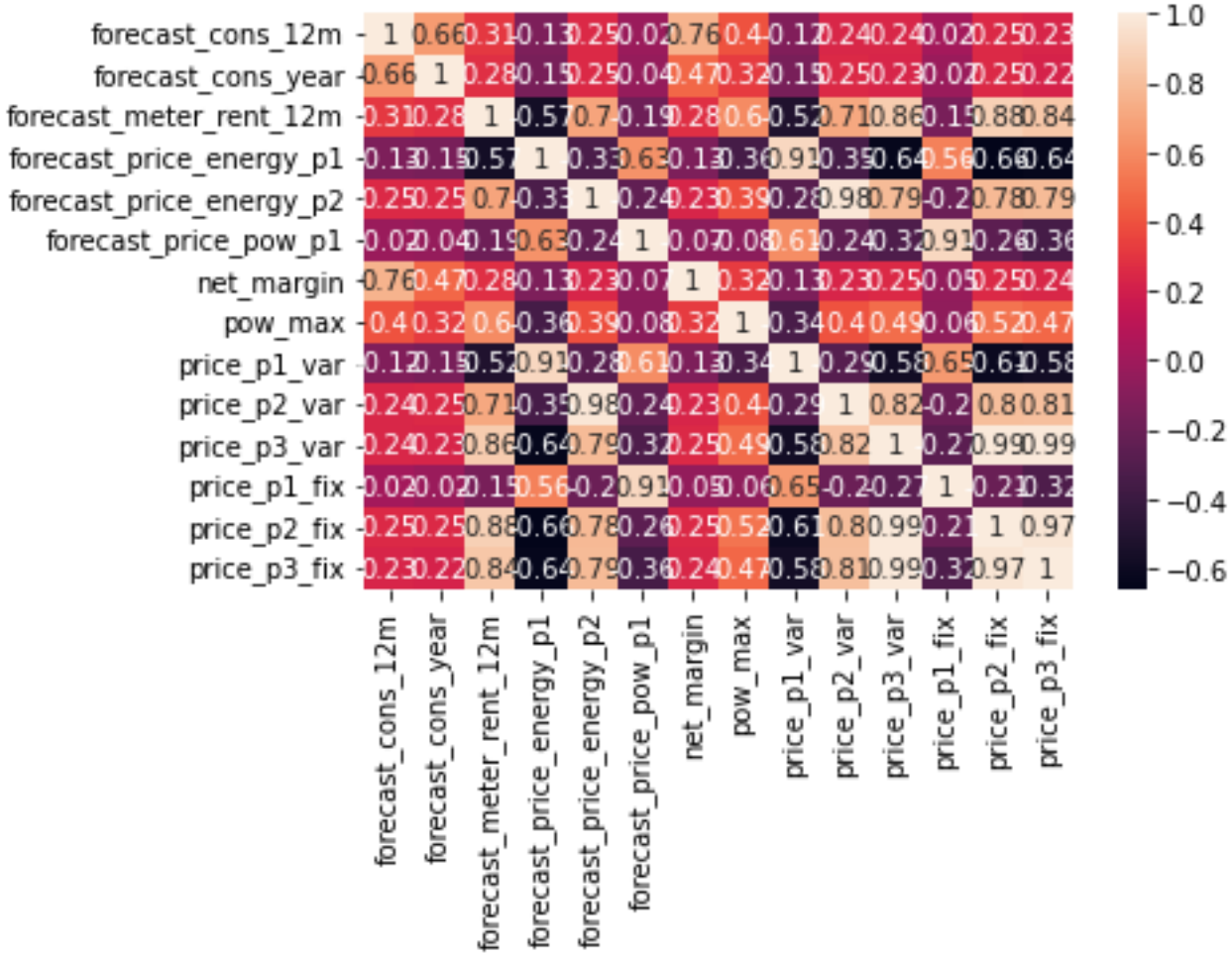


Figura 8: mappa di calore di correlazione con variabili più correlate con la variabile target

Infine è stata applicata l'Analisi a Componenti Principali, anche detta PCA.
La PCA è una procedura statistica che utilizza una trasformazione ortogonale. Essa converte un gruppo di variabili correlate in un gruppo di variabili non correlate e può essere utilizzata per esaminare le relazioni tra un gruppo di variabili. Quindi, proprio come in questo studio, può essere utilizzata per la riduzione della dimensionalità *(Reddy, G. T., Reddy, M., et al., 2020).*

Si supponga che un insieme di dati *x(1), x(2),.....xm* abbia *n* dimensioni. I dati a *n* dimensioni devono essere ridotti a *k* dimensioni *(k<< n)* utilizzando la PCA. La PCA è descritta di seguito:

1)Standardizzazione dei dati grezzi: I dati grezzi devono avere varianza unitaria e media zero.

$$x_j^i = \frac{x_j^i - \bar{x}_j}{\sigma_j} \quad \forall j$$

2) Calcolare la matrice di co-varianza dei dati grezzi come segue:

$$\sum = \frac{1}{m}\sum_i^m (x_i)(x_i)^T, \sum \in R^{n*n}$$

3) Calcolare l'autovettore e l'autovalore della matrice di co-varianza, come indicato nell'equazione 1.

$$u^T \sum = \lambda\mu$$
$$U = \begin{bmatrix} | & | & | \\ u_1 & u_2 \dots & u_n \\ | & | & | \end{bmatrix}, \quad u_i \in R^n \quad (1)$$

4) I dati grezzi devono essere proiettati in un sottospazio k-dimensionale: vengono scelti i primi k autovettori della matrice di co-varianza. Questi saranno la nuova base originale per i dati. Il calcolo del vettore corrispondente è riportato nell'equazione 2.

$$x_i^{new} = \begin{bmatrix} u_1^T x^i \\ u_2^T x^i \\ ::::: \\ ::::: \\ u_k^T x^i \end{bmatrix} \in R^k \qquad (2)$$

In questo modo, se i dati grezzi sono di n dimensioni, saranno ridotti a una nuova rappresentazione k dimensionale dei dati.

In particolare, nello studio svolto con la funzione *Standard Scaler*, sono stati standardizzati i dati poiché senza questa operazione preventiva la tecnica PCA non è in grado di trovare le componenti principali ottimali. Standardizzare le caratteristiche è importante quando confrontiamo misurazioni che hanno unità diverse; questa funzione consiste nel centrare la variabile su zero e standardizzare la varianza su 1. Le variabili misurate a scale diverse non contribuiscono allo stesso modo all'analisi e potrebbero finire per creare un *bias*. Per questo

motivo, la tecnica denominata *StandardScaler* presente all'interno del pacchetto sci-kit-learn rimuove la media e ridimensiona i dati in base alla varianza dell'unità permettendo di effettuare la standardizzazione necessaria per procedere con l'analisi a componenti principali.

Dopo aver standardizzato i dati è stata effettuata l'analisi a componenti principali e sono state ottenute complessivamente 28 componenti che sono raffigurate nel grafico sottostante con la relativa varianza spiegata

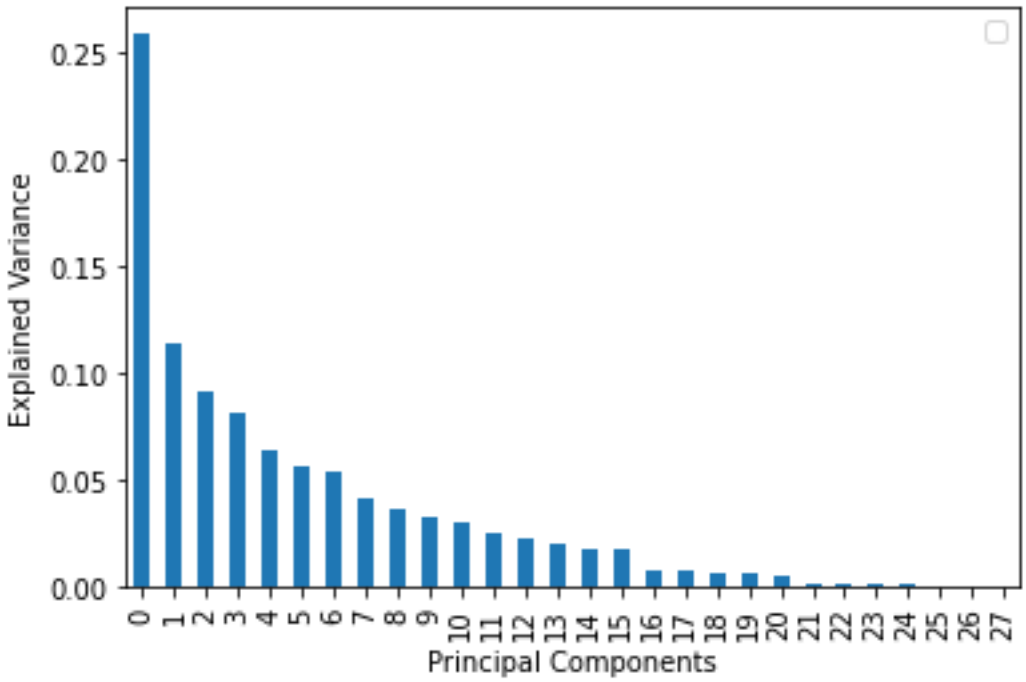
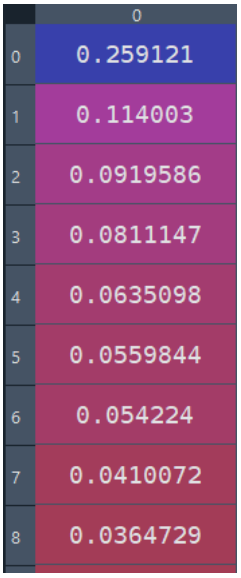


Figura 9:28 componenti principali con la relativa varianza spiegata

Successivamente, al fine di preservare un valore alto di varianza, è stato stabilito un valore fisso pari al 95% e il modello ha restituito 16 componenti principali. Dalle 28 componenti iniziali il dataset è stato ridotto quindi a 16 componenti mantendendo comunque alto il livello di informazione fornito dai dati.

Inoltre, con il comando explained_variance_ratio è stato possibile vedere che la prima componente principale contiene il 26% della varianza, mentre le restanti componenti principali hanno valori decisamente più bassi.



	0
0	0.259121
1	0.114003
2	0.0919586
3	0.0811147
4	0.0635098
5	0.0559844
6	0.054224
7	0.0410072
8	0.0364729

Figura 10: le prime 9 componenti principali con varianza spiegata

3.2.4. Estrazione delle features:

Le diverse tecniche ad oggi disponibili si differenziano nel modo sia in cui cercano il sottoinsieme più caratteristico, sia in cui incorporano il modello di classificazione (Saeys, et al., 2007). In questo lavoro sono state confrontate le due tecniche

La fase di selezione delle caratteristiche è un processo molto importante dell'analisi nel data mining, poiché è la fase in cui vengono determinate le caratteristiche critiche. La selezione delle caratteristiche non solo rimuove quelle indesiderate, ma ci aiuta anche a trovare quelle più rilevanti che permettono di aumentare le prestazioni del nostro modello.

In generale, le ragioni per cui si ricorre alla selezione delle caratteristiche sono:

- Ridurre il numero di caratteristiche, per ridurre l'overfitting e migliorare la generalizzazione dei modelli.

- Comprendere meglio le caratteristiche e la loro relazione con le variabili di risposta.

- Migliorare le prestazioni del modello e permettere la costruzione di modelli più veloci e meno costosi a livello computazionale.

- Ottenere una più profonda comprensione dei processi che sottostanno alla generazione dei dati.

Le diverse tecniche ad oggi disponibili si differenziano nel modo sia in cui cercano il sottoinsieme più caratteristico, sia in cui incorporano il modello di classificazione *(Saeys, et al., 2007).* In questo lavoro sono state confrontate due tecniche di selezione delle caratteristiche univariate; le tecniche univariate esaminano ogni caratteristica singolarmente per determinare la forza della relazione della caratteristica con la variabile di risposta.

Esse sono rispettivamente:

1. Tecnica Anova,
2. Tecnica di Mutua informazione.

➢ Tecnica Anova:

L'analisi della varianza (ANOVA) è un metodo statistico utilizzato per verificare le medie di due o più gruppi che sono significativamente diverse tra loro.

L'ANOVA utilizza il test F per verificare se esiste una differenza significativa tra le medie. Se non c'è una differenza significativa tra le medie e se tutte le varianze sono uguali, il risultato del rapporto F dell'ANOVA sarà prossimo a 1.

Nello studio condotto applicando questa tecnica sono stati ottenuti i seguenti valori per le singole variabili:

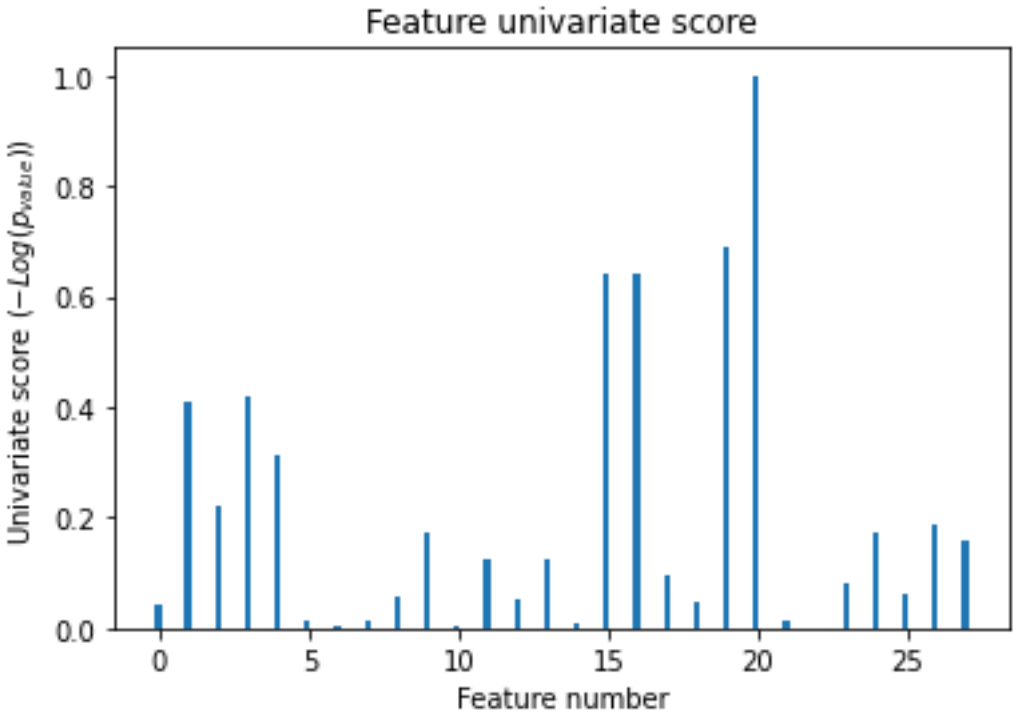


Figura 11: punteggio univariato per ciascuna variabile

Dal grafico emerge che, nel set totale di caratteristiche, solo 4 caratteristiche sono significative.

➢ Tecnica di Mutua Informazione:

La Mutual Information (MI) è un metodo di filtraggio univariato che fornisce una migliore accuratezza al modello. I metodi di filtraggio univariati sono abbastanza veloci e possono essere utilizzati come screening per ottenere una migliore accuratezza e un minor tempo di

addestramento. Come tutte le altre tecniche di selezione delle caratteristiche, anche questa mira a ridurre la dimensione dell'insieme di caratteristiche in ingresso e, allo stesso tempo, a conservare le informazioni discriminanti della classe per i problemi di classificazione. La riduzione delle caratteristiche può ridurre la complessità del problema o il suo tempo di calcolo e persino portare a un miglioramento dell'accuratezza del modello.

La mutua informazione (MI) tra due variabili casuali è un valore non negativo, che misura la dipendenza tra le variabili. È uguale a zero se e solo se due variabili casuali sono indipendenti, mentre valori più alti significano una maggiore dipendenza.

Si può sintetizzare, quindi, come la quantità di informazioni che una variabile fornisce all'altra.

L'informazione reciproca tra due variabili casuali X e Y può essere espressa formalmente come segue:

$$I(X ; Y) = H(X) - H(X \mid Y)$$

Dove $I(X; Y)$ è l'informazione reciproca per X e Y, $H(X)$ è l'entropia per X e $H(X \mid Y)$ è l'entropia condizionale per X dato Y.

La mutua informazione è una misura di dipendenza o "dipendenza reciproca" tra due variabili casuali. In quanto tale, la misura è simmetrica, cioè

$$I(X; Y) = I(Y; X).$$

L'entropia quantifica la quantità di informazione presente in una variabile casuale. L'informazione reciproca aiuta quindi a ridurre l'entropia.

Nello studio condotto

Dal grafico, possiamo dedurre che “forecast_cons_12m” ha il più alto guadagno di informazioni reciproche (0,67), poi cons_12m (0,65), seguito da net_margin (0,64), e così via.

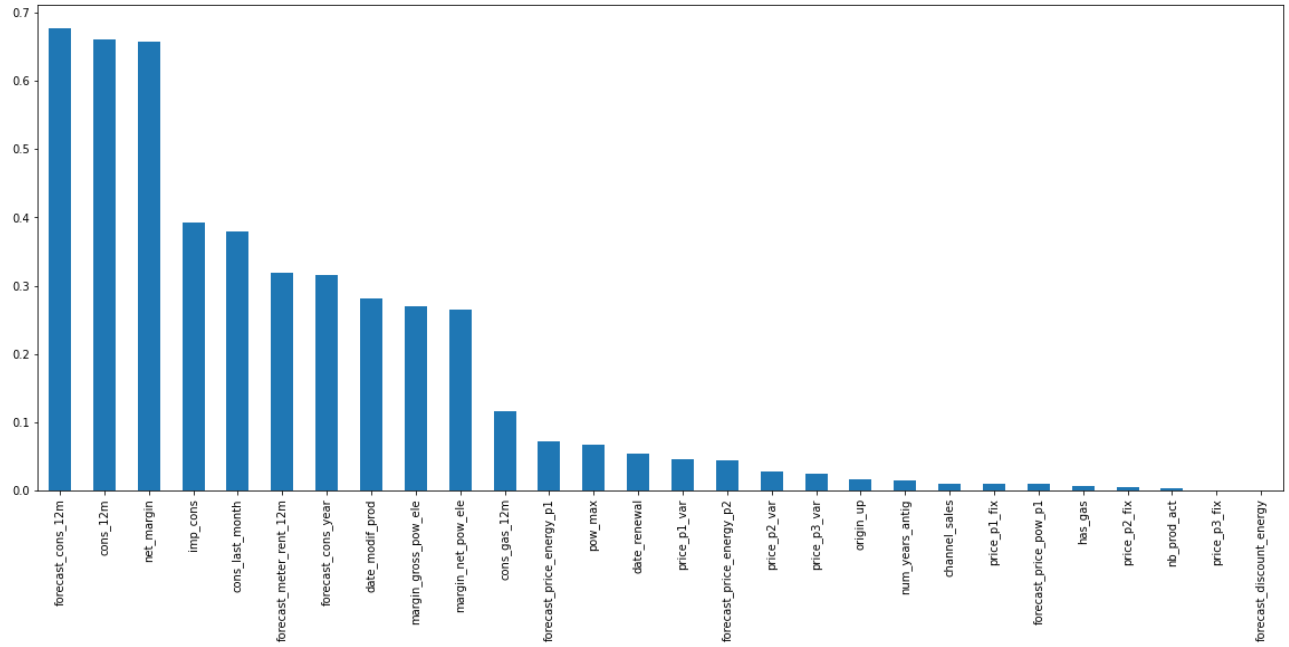


Figura 12: Mutual Information per ciascuna variabile del dataset

Questa tecnica ha permesso inoltre di identificare le variabili meno rilevanti per lo studio e che quindi sono state eliminate per aumentare l'accuratezza dei modelli implementati.

Queste variabili sono: "forecast_discount_energy", "price_p3_fix", "nb_prod_act", "price_p2_fix" che, come possiamo vedere dalla tabella rappresentata, hanno valori inferiori a 0,001.

index	0
channel_sales	0.011624
cons_12m	0.660595
cons_gas_12m	0.115604
cons_last_month	0.378052
date_modif_prod	0.281907
date_renewal	0.0574655
forecast_cons_12m	0.677604
forecast_cons_year	0.31939
forecast_discount_energy	0
forecast_meter_rent_12m	0.317962
forecast_price_energy_p1	0.0730307
forecast_price_energy_p2	0.0493024
forecast_price_pow_p1	0.0132212
has_gas	0
imp_cons	0.390255
margin_gross_pow_ele	0.269913
margin_net_pow_ele	0.271782
nb_prod_act	0.00218876
net_margin	0.654803
net_margin	0.654803
num_years_antig	0.010318
origin_up	0.0135051
pow_max	0.0737928
price_p1_var	0.0452402
price_p2_var	0.0267002
price_p3_var	0.0243175
price_p1_fix	0.00675748
price_p2_fix	0.00499212
price_p3_fix	0.00632127

Figura 13: valore mutua informazione per ciascuna variabile

Uno degli aspetti chiave dell'apprendimento automatico supervisionato è la valutazione e la convalida dei modelli. Quando si valutano le prestazioni predittive del modello, è essenziale che il processo sia imparziale. Utilizzando train_test_split() della libreria dati *scikit-learn*, è possibile dividere il set di dati in sottoinsiemi che riducono al minimo il potenziale di distorsione nel processo di valutazione e validazione.

Per valutare le prestazioni predittive del modello e convalidarlo è necessaria, quindi, una valutazione imparziale. Ciò significa che non è possibile valutare le prestazioni predittive di un modello con gli stessi dati utilizzati per l'addestramento. È necessario valutare il modello con dati nuovi che il modello non ha mai visto prima. È possibile ottenere questo risultato dividendo il set di dati prima di utilizzarlo.

Proprio come in questo caso, il set di dati viene suddiviso in tre sottoinsiemi:

1. Il *training set*: che, come si deduce dal nome stesso, viene utilizzato per addestrare, o adattare, il modello.
2. Il *validation set*: viene utilizzato per la valutazione del modello durante la regolazione degli iperparametri.
3. Il *test set:* serve a testare il modello su un nuovo set di dati che non conosce. Permette quindi di valutare la validità del modello finale in modo del tutto imparziale.

In alcuni casi, quando non è necessario sintonizzare gli iperparametri, è possibile lavorare solo con il train set e il test set.

In particolare, nello studio svolto il dataset è stato ripartito rispettivamente 80% in *train* e 20% in *test.*

3.2.5. Scelta del modello e metriche:

In questa fase conclusiva sono stati allenati gli algoritmi per risolvere il task di classificazione, sono state analizzate le metriche per valutare le performance dei diversi modelli allenati e infine è stato scelto il modello che, per questo tipo di task, risulta essere più performante.

Gli algoritmi di classificazione nell'apprendimento automatico contengono diversi algoritmi, e in questo lavoro, il documento si è concentrato principalmente sul random forest, il decision tree, il knn e il Naive Bayes.

- Random Forest:

Il primo modello che è stato applicato allo studio è il *Random Forest*. Il random forest è un algoritmo statistico o di apprendimento automatico per la predizione.

Prima ancora però di analizzare il random forest è necessario fare un'introduzione preventiva sui modelli *Decision Tree* poichè rappresentano gli elementi costitutivi dell'algoritmo random forest.
Un modello ad albero prevede la partizione ricorsiva del dataset dato in due gruppi in base a un determinato criterio fino a quando non viene soddisfatta una condizione di arresto predeterminata. Alla base degli alberi decisionali si trovano i cosiddetti nodi foglia o foglie.

La Figura 14 illustra un partizionamento ricorsivo di uno spazio di input bidimensionale con confini allineati agli assi, cioè ogni volta lo spazio di input viene partizionato in una direzione parallela a uno degli assi. Qui la prima suddivisione è avvenuta su $x2 \geq a2$. Poi, i due sottospazi sono stati nuovamente partizionati: Il ramo sinistro è stato diviso su $x1 \geq a4$. Il ramo destro è stato prima diviso su $x1 \geq a1$ e uno dei suoi sotto rami è stato diviso su $x2 > a3$.

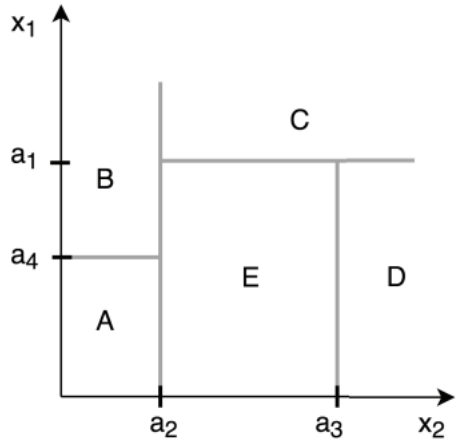


Figura 14: Partizione binaria ricorsiva di un sottospazio bidimensionale (Schonlau, M., Zou, R. Y., 2020)

La figura successiva, invece, rappresenta graficamente i sottospazi suddivisi nella figura 1.

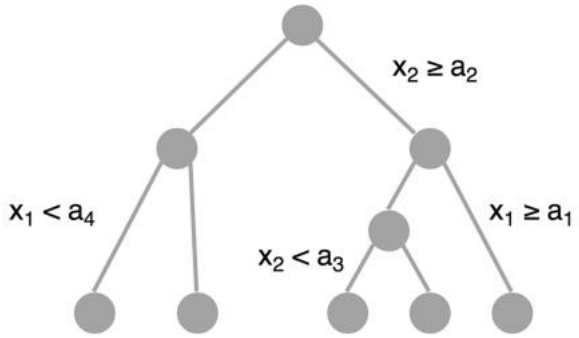


Figura 15: Rappresentazione grafica dell'albero decisionale di cui alla figura 1 (Schonlau, M., Zou, R. Y., 2020)

A seconda di come vengono impostati i criteri di partizione e di arresto, gli alberi decisionali possono essere progettati sia per compiti di classificazione sia per compiti di regressione. Sia per i problemi di classificazione che per quelli di regressione, il sottoinsieme delle variabili predittive selezionate per suddividere un nodo interno dipende da criteri di suddivisione predefiniti, formulati come un problema di ottimizzazione. Un criterio di suddivisione comune nei problemi di classificazione è l'entropia, che consiste nell'applicazione pratica del teorema di Shannon sulla codifica delle fonti, che specifica il limite inferiore della lunghezza della

rappresentazione in bit di una variabile casuale. In ogni nodo interno dell'albero decisionale, l'entropia è data dalla seguente formula (Schonlau, M., Zou, R. Y., 2020):

$$E = -\sum_{i=1}^{c} p_i \times \log(p_i)$$

Dove *c* rappresenta il numero di classi uniche e *pi* è la probabilità preventiva di ogni classe. Questo valore viene massimizzato per ottenere il massimo delle informazioni a ogni divisione dell'albero decisionale.

➢ Decision Tree:

I Decision Tree (DT) sono considerati uno dei metodi più noti per la rappresentazione della classificazione dei dati da parte dei classificatori. Il DT è un modello di classificazione utilizzato nel Data Mining. L'obiettivo è creare un modello che preveda il valore di una variabile target in base a diverse variabili di input. Per quanto riguarda la struttura, come anticipato, ciascun albero è composto da nodi e rami. Ogni nodo rappresenta caratteristiche di una categoria da classificare e ogni sottoinsieme definisce un valore che può essere assunto dal nodo. A causa della loro semplice analisi e della loro precisione su più forme di dati, gli alberi decisionali hanno trovato molti campi di applicazione *(Jijo, B. T., Abdulazeez, A. M., 2021).*

Un albero di decisione è un albero in cui ogni nodo interno è etichettato con una caratteristica di input. Gli archi provenienti da un nodo etichettato con una caratteristica di input sono etichettati con ciascuno dei possibili valori della caratteristica target o l'arco conduce a un nodo decisionale subordinato su una diversa caratteristica di input. Ogni foglia dell'albero è etichettata con una classe o con una distribuzione di probabilità sulle classi, a significare che l'insieme dei dati è stato classificato dall'albero in una classe specifica o in una particolare distribuzione di probabilità.

Nel data mining, gli alberi decisionali possono essere descritti anche come la combinazione di tecniche matematiche e computazionali per aiutare la descrizione, la categorizzazione e la generalizzazione di un dato insieme di dati.

I dati si presentano sotto forma di:

$$(\mathbf{x}, Y) = (x_1, x_2, x_3, \ldots, x_k, Y)$$

Dove la variabile dipendente Y, è la variabile target da classificare. Il vettore x invece è composto dalle caratteristiche *x1, x2, x3*, …, xk che vengono utilizzate per il task di classificazione *(Jijo, B. T., Abdulazeez, A. M., 2021)*.

➢ Naïve – Bayes:

il classificatore *Naïve Bayes* (NB) è un algoritmo di apprendimento automatico supervisionato, utilizzato per compiti di classificazione. Fa parte di una famiglia di algoritmi di apprendimento generativo, ovvero cerca di modellare la distribuzione degli input di una determinata classe o categoria. Naïve Bayes è noto anche come classificatore probabilistico, poiché si basa sul teorema di Bayes.

Il teorema di Bayes, noto anche come Regola di Bayes, ci permette di "invertire" le probabilità condizionali. Le probabilità condizionali rappresentano la probabilità di un evento al verificarsi di un altro evento, che viene rappresentato con la seguente formula:

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

In questo caso, A e B sono due eventi, P(A) e P(B) sono le due probabilità di A e B se trattati come eventi indipendenti, e P(A|B) e P(B|A) sono le probabilità composte di A dato B e B dato A, rispettivamente.

Il Teorema di Bayes si distingue per l'uso di eventi sequenziali, in cui le informazioni aggiuntive acquisite successivamente influiscono sulla probabilità iniziale. Queste probabilità sono denominate probabilità anteriore e probabilità posteriore. La probabilità anteriore è la probabilità iniziale di un evento prima che sia contestualizzato in una certa condizione, o la probabilità marginale. La probabilità posteriore è la probabilità di un evento dopo l'osservazione di un dato.

Il classificatore bayesiano richiede la conoscenza delle probabilità a priori e condizionali relative al problema, quantità che in generale non sono note ma sono tipicamente stimabili. Se è

possibile ottenere delle stime affidabili delle probabilità coinvolte nel teorema, il classificatore bayesiano risulta generalmente affidabile e potenzialmente compatto.

➢ K-Nearest Neighbors:

L'algoritmo *k-nearest neighbors*, noto anche come KNN o k-NN, è un classificatore non parametrico di apprendimento supervisionato che utilizza la prossimità per effettuare classificazioni o previsioni sul raggruppamento di un singolo punto di dati. Sebbene possa essere utilizzato sia per problemi di regressione che di classificazione, è tipicamente usato come algoritmo di classificazione, partendo dal presupposto che i punti simili possono essere trovati l'uno vicino all'altro.

L'algoritmo kNN può essere considerato un sistema di valutazione, in cui l'etichetta di classe maggioritaria determina l'etichetta di un nuovo punto di dati tra i suoi 'k' (dove k è un numero intero) più vicini nello spazio delle caratteristiche. dove l'etichetta di classe maggioritaria determina l'etichetta di classe di un nuovo punto dati tra i suoi k vicini.

Di default, il Knn per calcolare la distanza tra le varie etichette utilizza la distanza euclidea, che può essere calcolata con la seguente equazione:

$$D(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2}$$

dove p e q sono confrontati con n caratteristiche.

Un altro concetto è il parametro k, che decide il numero di vicini da scegliere per l'algoritmo kNN. La scelta appropriata di k ha un impatto significativo sulle prestazioni diagnostiche dell'algoritmo kNN. Un k grande riduce l'impatto della varianza causata dall'errore casuale, ma corre il rischio di ignorare modelli piccoli ma importanti. La chiave per scegliere un valore k appropriato è trovare un equilibrio tra l'overfitting e l'underfitting *(Zhang, Z., 2016).*

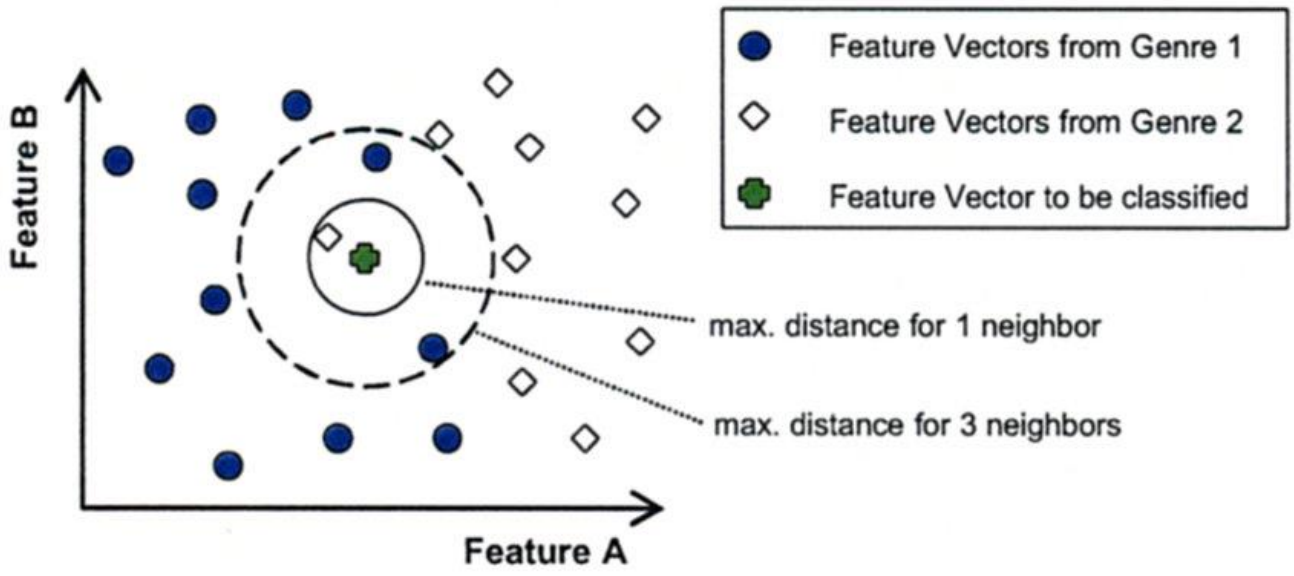


Figura 16: algoritmo K – Nearest Neighbor (researchgate.net)

3.2.5 Metriche:

Esistono diverse metriche che possono essere utilizzate per valutare le prestazioni degli algoritmi di Machine Learning (ML).

In questo studio sono state analizzate principalmente:

- La matrice di confusione
- Accuracy
- Precision
- Recall
- F1- score
- Curva ROC
- Curva AUC

La scelta di confrontare più metriche nello studio permette di aumentare l'attendibilità dell'analisi svolta.
Di seguito viene riportata una breve spiegazione delle metriche analizzate:

La matrice di confusione è una delle metriche più intuitive e più semplici utilizzate per trovare la correttezza e l'accuratezza del modello. Viene utilizzata per problemi di classificazione come in questo caso in cui l'output è composto da due classi (0 = no churn, 1 = si churn).
Una matrice di confusione è una matrice NxN, dove N è il numero di classi previste. Per il problema in esame, dato che sono presenti 2 classi N sarà uguale a 2 e quindi si otterrà una matrice 2x2. La matrice di confusione è una tabella con due dimensioni ("Effettiva" e "Prevista") e un insieme di "classi" in entrambe le dimensioni *(Dalianis, H., Dalianis, H., 2018).*

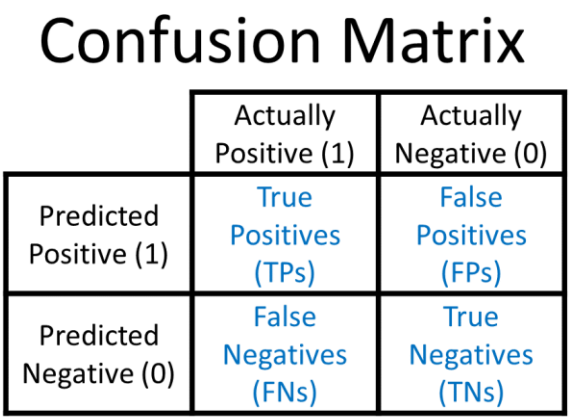


Figura 17: Matrice di confusione (glassboxmedicine.com)

La diagonale centrale all'interno della matrice rappresenta le classi che sono state classificate correttamente (TN e TP), mentre i riquadri esterni rappresentano gli errori del modello (FN e FP).
Più precisamente:

True Negatives (TN) – si verifica quando la previsione è 0 e la vera classe è effettivamente 0, cioè il modello prevede correttamente che la classe è negativa (0).

False Positive (FP) – si verifica quando la previsione è 0 mentre la vera classe è in realtà 1, ovvero il modello prevede erroneamente che la classe è negativa (0).

False Negative (FN) – si verifica quando la previsione è 1 mentre la vera classe è in realtà 0, ovvero il modello prevede erroneamente che la classe è positiva (1).

True Positive (TP) – si verifica quando la previsione è 1 mentre la vera classe è in realtà 1, cioè il modello prevede correttamente che la classe è positiva (1).

Per quanto riguarda le altre metriche analizzate possiamo affermare che:

Accuracy: L'accuratezza è il grado di corrispondenza del dato teorico, desumibile da una serie di valori misurati (campione di dati), con il dato reale, ovvero la differenza tra valore medio campionario e valore vero.

$$Accuracy : A = \frac{tp + tn}{tp + tn + fp + fn}$$

Precision : La precision misura il numero di istanze corrette rilevate diviso per tutte le istanze rilevate.

$$Precision : P = \frac{tp}{tp + fp}$$

Recall: La recall misura il numero di istanze corrette rilevate diviso per tutte le istanze corrette,

$$Recall : R = \frac{tp}{tp + fn}$$

F-score: è definito come la media ponderata di precisione e richiamo a seconda della funzione di peso β

$$F\text{-}score : F_\beta = (1 + \beta^2) * \frac{P * R}{\beta^2 * P + R}$$

Con β = 1 si ottiene il punteggio standard F

$$F\text{-}score : F_1 = F = 2 * \frac{P * R}{P + R}$$

F1-score: indica la media armonica tra precision e recall.

Oltre alle metriche sono stati analizzati due grafici che permettono di valutare e confrontare le performance dei vari modelli. Essi sono la Curva Roc e la Curva Auc:

La Curva ROC (Receiver Operator Characteristics) è un grafico utilizzato per mostrare la capacità diagnostica dei classificatori binari. Essa viene costruita tracciando il tasso di veri positivi (TPR) rispetto al tasso di falsi positivi (FPR). Il tasso di veri positivi è la percentuale di osservazioni correttamente previste come positive su tutte le osservazioni positive (TP/(TP + FN)). Allo stesso modo, il tasso di falsi positivi è la percentuale di osservazioni erroneamente previste come positive su tutte le osservazioni negative (FP/(TN + FP)).

La curva ROC mostra il compromesso tra sensibilità (o TPR) e specificità (1 - FPR). I classificatori che forniscono curve più vicine all'angolo in alto a sinistra indicano una migliore performance.

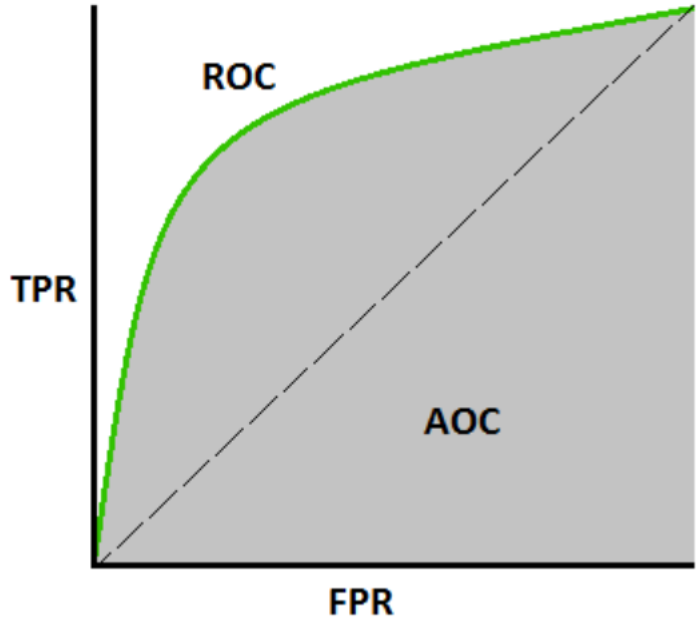


Figura 18: Curva ROC e AUC, (Narkhede, S., 2018)

La Curva AUC (Area Under the Curve), è la misura della capacità di un classificatore binario di distinguere tra le classi. Essa misura l'intera area bidimensionale sotto l'intera curva ROC da (0,0) a (1,1). È importante affermare che, più alto è l'AUC, migliore è la performance del modello nel distinguere tra classi positive e negative.

In particolare, quando il valore dell'AUC = 1, il classificatore è in grado di distinguere correttamente tra tutti i punti della classe Positivi e quelli Negativi. Se invece l'AUC fosse stata pari a 0, il classificatore avrebbe completamente confuso le classi prevedendo tutti i negativi come positivi e tutti i positivi come negativi. Quando l'AUC=0,5, il classificatore non è in grado di distinguere tra punti di classe positivi e negativi. Ciò significa che il classificatore predice una classe casuale o una classe costante per tutti i punti dati.

3.3 Risultati e discussioni:

In questa sezione vengono esposti e confrontati i risultati ottenuti dai diversi modelli utilizzati nello studio. Nello studio complessivamente sono stati applicati quattro algoritmi che sono rispettivamente: Random Forest, Decision Tree, Naive Bayes e K-Neighrest Neighbors.

Il confronto è stato fatto mantenendo la distinzione tra i risultati ottenuti con le due tecniche di estrazione delle caratteristiche; rispettivamente la tecnica Anova e la tecnica Mutual Information.

➢ Random forest:

Dall'analisi delle metriche emerge che l'algoritmo Random forest ha performato particolarmente bene sia applicando la tecnica Anova sia la Mutual Information. Come possiamo vedere dalle matrici di confusione rappresentate risulta che il modello ha classificato correttamente entrambe le classi 0 e 1. La diagonale che va da in alto a sinistra a in basso a destra, infatti, indica quante previsioni abbiamo ottenuto esattamente. L'altra diagonale rappresenta il nostro numero di errori di tipo zero e 1.

Nel complesso il maggior numero di errori si trova nel riquadro in basso a sinistra della matrice di confusione. Sono stati individuati rispettivamente 66 Falsi Negativi con la tecnica Anova e 38 Falsi Negativi con la tecnica Mutual Information.

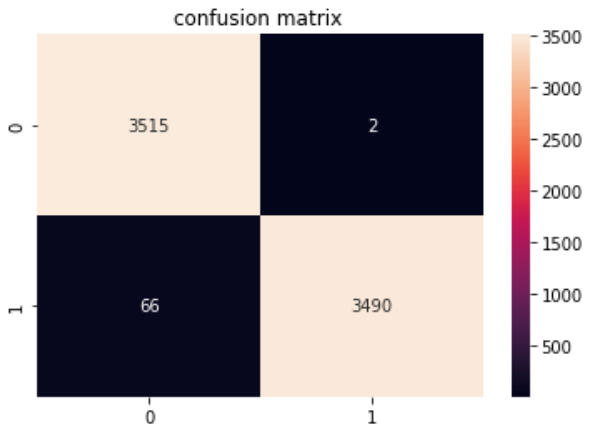


Figura 19: Matrice di confusione con tecnica Anova

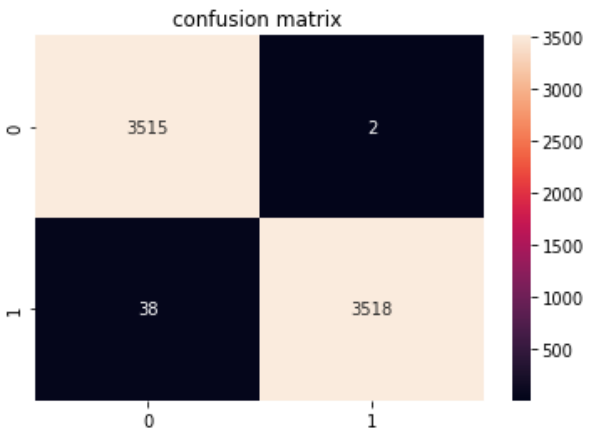


Figura 20: Matrice di confusione con tecnica MI

Nel report di classificazione sono riassunte le principali metriche calcolate quali: precision, recall, l'F1-score e l'accuracy. Complessivamente le metriche confermano che il modello ha performato bene e, seppur con una leggerissima differenza, il random forest ha performato meglio con la tecnica Mutual Information.

	precision	recall	f1-score	support
0	0.98	1.00	0.99	3517
1	1.00	0.98	0.99	3556
accuracy			0.99	7073
macro avg	0.99	0.99	0.99	7073
weighted avg	0.99	0.99	0.99	7073

Figura 21: Report di classificazione con tecnica Anova

	precision	recall	f1-score	support
0	0.99	1.00	0.99	3517
1	1.00	0.99	0.99	3556
accuracy			0.99	7073
macro avg	0.99	0.99	0.99	7073
weighted avg	0.99	0.99	0.99	7073

Figura 22: Report di classificazione con tecnica MI

La Curva ROC in entrambi gli scenari proposti è prossima al valore 1. Ciò significa che il test è altamente accurato e quindi il test ha discriminato correttamente i clienti che hanno abbandonato il servizio dai clienti che non hanno abbandonato il servizio.

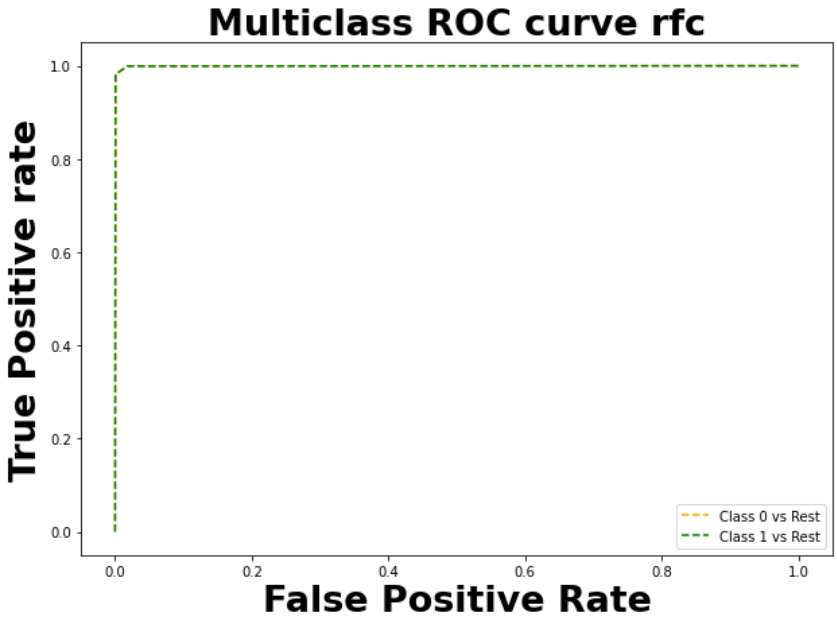


Figura 23: Curva ROC Random forest

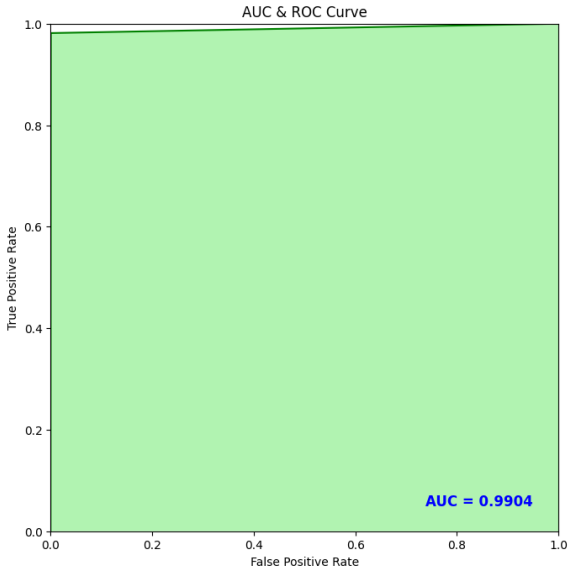


Figura 24: Curva AUC con tecnica Anova

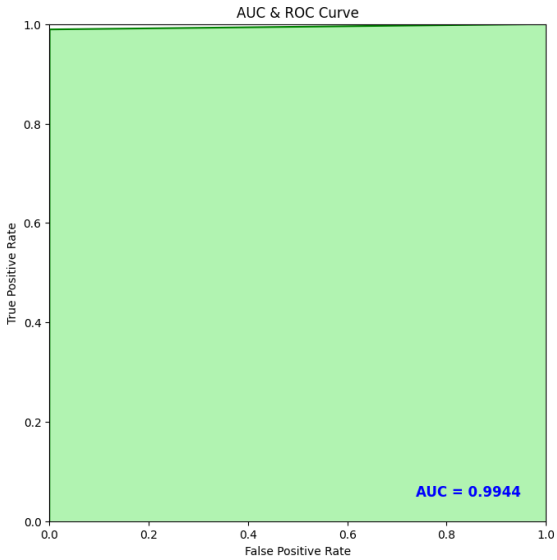


Figura 25: Curva AUC con tecnica MI

Anche la Curva AUC conferma quanto è stato affermato fino ad ora. Il modello con la tecnica Anova fornice un valore pari a 0,9904 mentre con la tecnica MI fornisce un risultato leggermente più alto pari a 0,9944.

➢ Decision Tree:

Analogamente il Decision Tree, che si presta molto bene ai problemi di classificazione, ha raggiunto risultati molto buoni. Complessivamente si può affermare che è il modello che ha performato meglio tra quelli analizzati fin'ora nello studio.

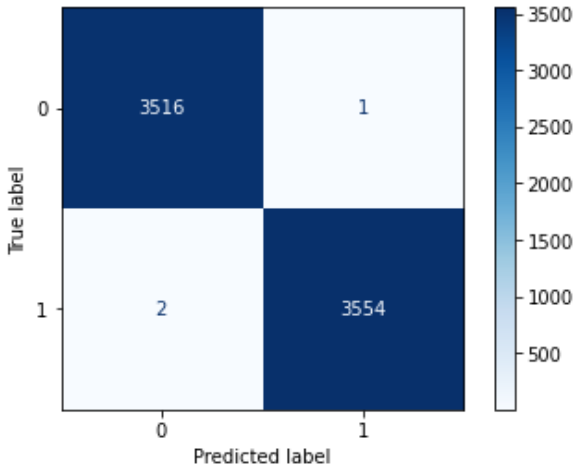


Figura 26: Matrice di confusione con tecnica Anova

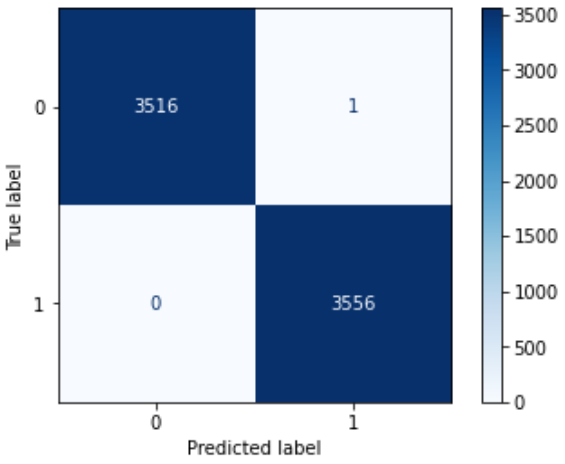


Figura 27: Matrice di confusione con tecnica MI

Dall'analisi delle matrici di confusione emerge infatti che la quasi totalità dei dati analizzati è stata classificata correttamente con entrambe le tecniche di feature selection (Anova e MI).

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3517
1	1.00	1.00	1.00	3556
accuracy			1.00	7073
macro avg	1.00	1.00	1.00	7073
weighted avg	1.00	1.00	1.00	7073

Figura 28: Report di classificazione Decision Tree

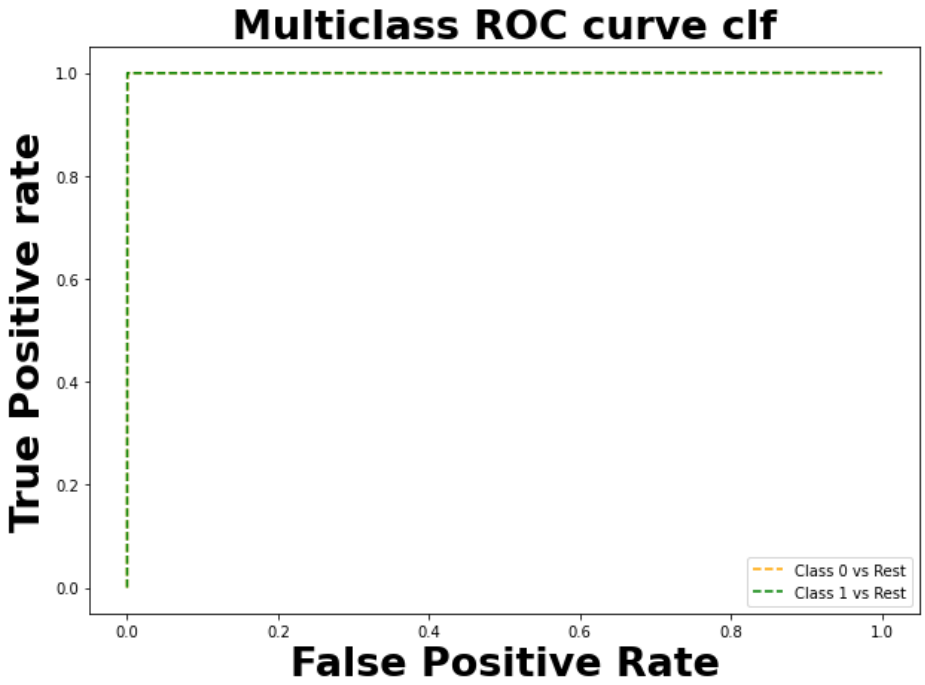


Figura 29: Curva ROC Decision Tree

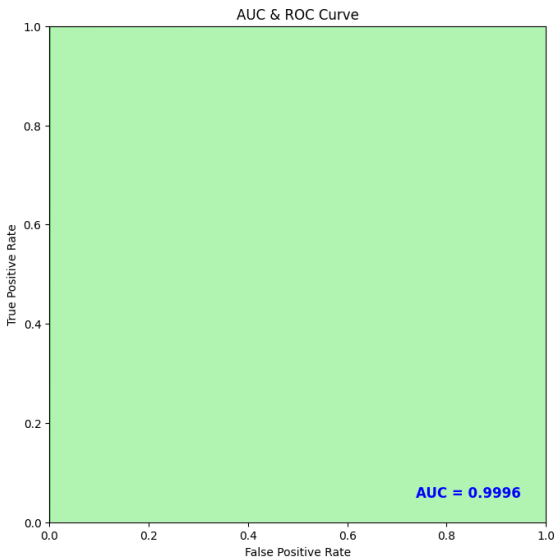


Figura 30: Curva AUC con tecnica Anova

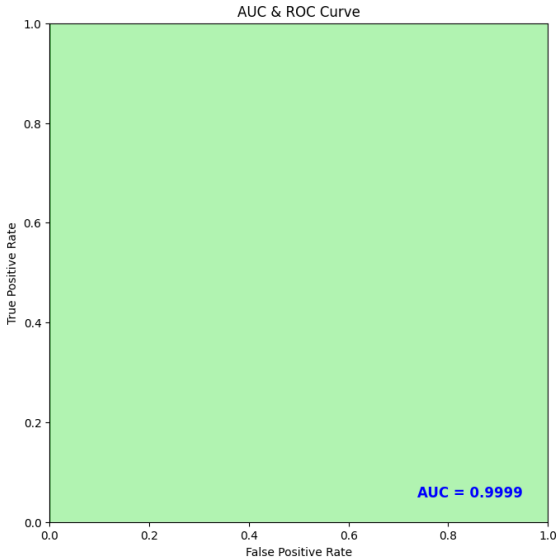


Figura 31: Curva AUC con tecnica MI

È possibile notare una leggera differenza dal valore fornito dalla curva AUC con le due tecniche adottate. Ottenendo un valore pari a 0,9996 con la tecnica Anova e 0,9999 con la tecnica Mutual Information.

➢ Naïve – Bayes:

A differenza dei due modelli precedentemente analizzati il modello Naive-Bayes non risulta dall'analisi delle metriche altrettanto performante.

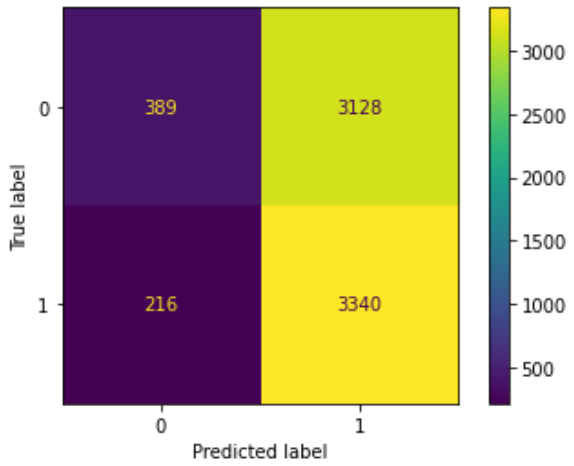


Figura 32: Matrice di confusione con tecnica Anova

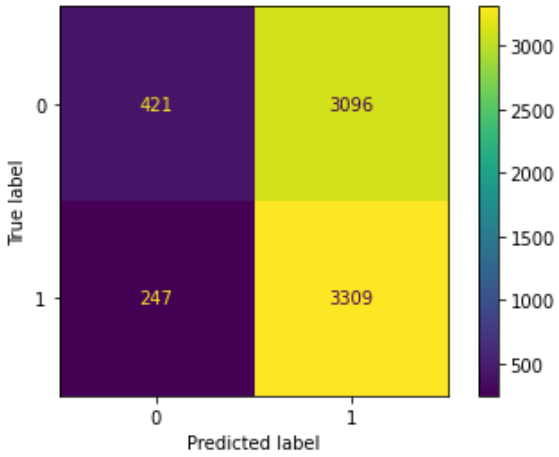


Figura 33: Matrice di confusione con tecnica MI

Come è possibile osservare dalle matrici di confusione l'algoritmo, applicando la tecnica Anova, classifica correttamente 389 valori come veri positivi e 3340 come veri negativi. Allo stesso tempo però classifica erroneamente 216 dati come falsi negativi e 3128 dati come falsi positivi.

La matrice di confusione restituita dal modello con la tecnica MI risulta leggermente più performante.

	precision	recall	f1-score	support
0	0.64	0.11	0.19	3517
1	0.52	0.94	0.67	3556
accuracy			0.53	7073
macro avg	0.58	0.52	0.43	7073
weighted avg	0.58	0.53	0.43	7073

Figura 34: Report di classificazione con tecnica Anova

	precision	recall	f1-score	support
0	0.63	0.12	0.20	3517
1	0.52	0.93	0.66	3556
accuracy			0.53	7073
macro avg	0.57	0.53	0.43	7073
weighted avg	0.57	0.53	0.43	7073

Figura 35: Report di classificazione con tecnica MI

Per quanto riguarda i report di correlazione, in entrambi gli scenari, restituiscono metriche con valori molto bassi. Il modello ottiene delle metriche leggermente migliori con la Mutual Information con una accuracy pari a 0,53 e un F1-score pari a 0,20.

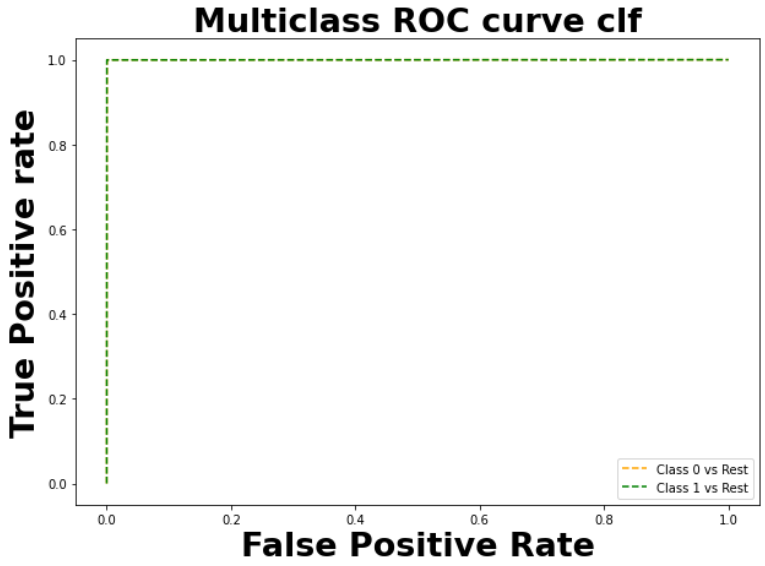


Figura 36: Curva ROC Naive Bayes

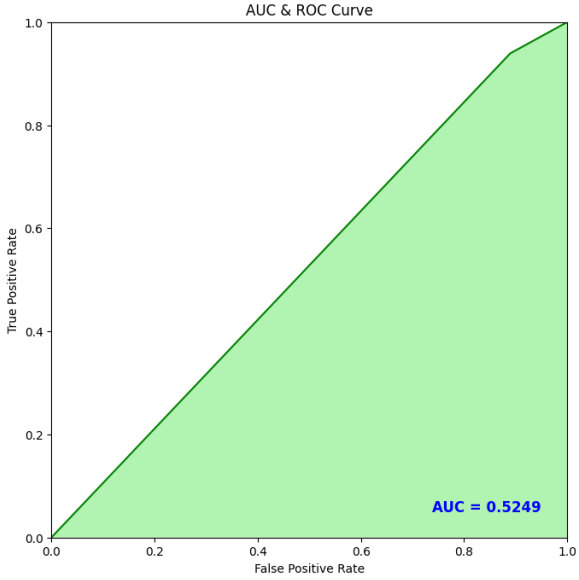


Figura 37: Curva AUC con tecnica Anova

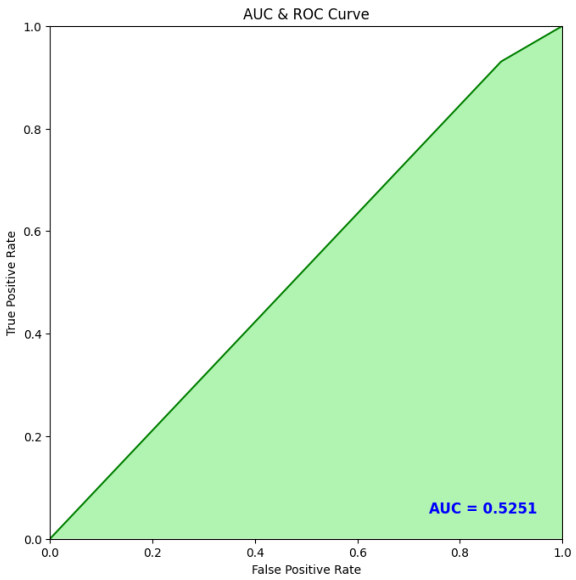


Figura 38: Curva AUC con tecnica MI

Anche dalla curva AUC si evince molto chiaramente che il modello non si presta per questo tipo di task. I grafici mostrano infatti valori AUC molto bassi pari a 0,5249 e 0,5251 applicando rispettivamente la tecnica Anova e la MI.

➢ K-Nearest Neighbors:

Dall'analisi delle metriche si desume che il modello K-Nearest Neighbors tra tutti sia il modello più performante per questo tipo di task.

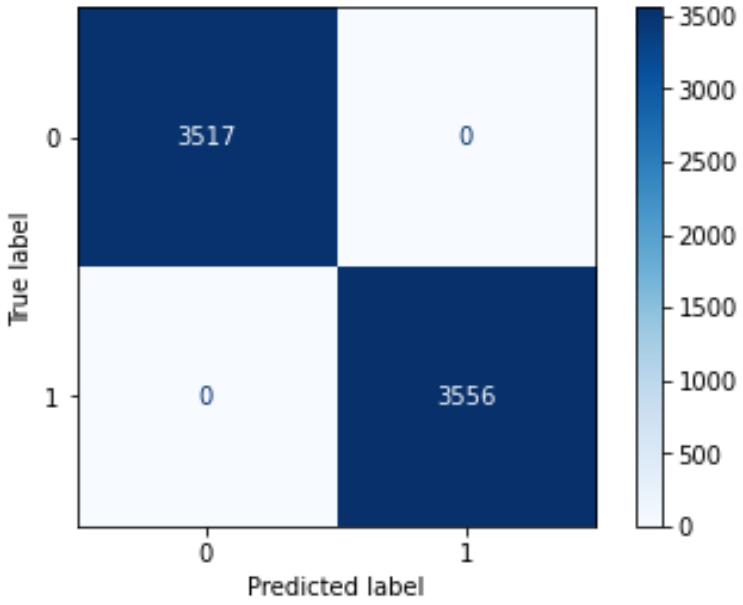


Figura 39: Matrice di confusione K-NN

Nella matrice di confusione si può notare facilmente come tutti i dati siano stati classificati correttamente come True Positive(s) e True Negative(s). I risultati ottenuti sono analoghi sia utilizzando la tecnica di feature selection Anova sia la MI.

	precision	recall	f1-score	support
0	1.00	1.00	1.00	3517
1	1.00	1.00	1.00	3556
accuracy			1.00	7073
macro avg	1.00	1.00	1.00	7073
weighted avg	1.00	1.00	1.00	7073

Figura 40: Report di classificazione K-NN

Allo stesso modo, il report di classificazione in entrambi i casi restituisce risultati molto alti con una accuracy e un F1-score pari a 1.

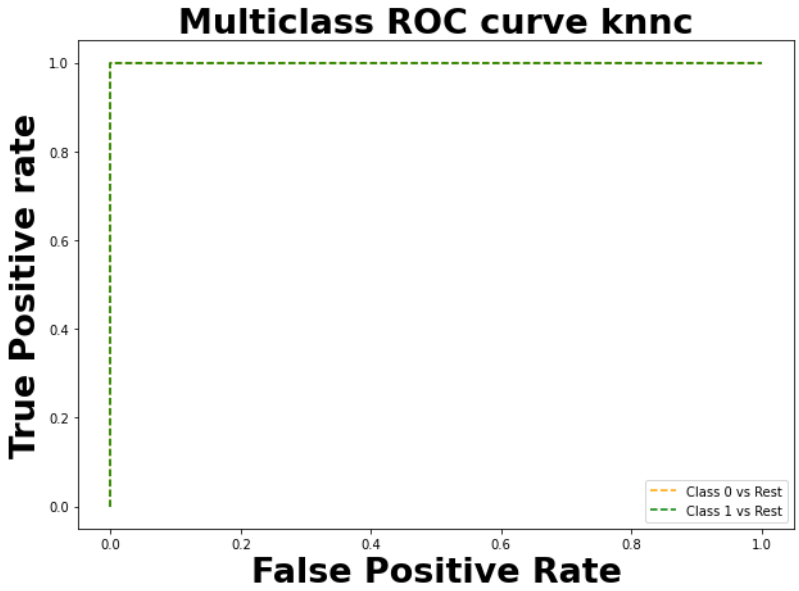


Figura 41: Curva ROC K-NN

4.Conclusioni e sviluppi futuri

In questa sezione viene fornita una spiegazione dell'algoritmo scelto e vengono descritti i futuri studi che potrebbero essere implementati sul medesimo set di dati.
Dal confronto dei diversi algoritmi di apprendimento si può affermare che il modello che si presta meglio per questo tipo di task è il K-Nearest Neighbors. È possibile affermare ciò a partire da un accurato confronto dei vari modelli di classificazione. Dal confronto e dall'analisi delle metriche, infatti, è emerso che il K - NN è il modello che ha ottenuto i valori più alti e che quindi ha performato meglio con entrambe le tecniche di features selection; rispettivamente con la tecnica Anova e la Mutual Information.
Nella *figura 39*, raffigurante la matrice di confusione, è possibile vedere come il modello abbia classificato correttamente tutte le istanze. All'interno della stessa matrice non è stato individuato nessun errore. Per questo motivo si può affermare che il classificatore K-NN si presta particolarmente bene per questo tipo di task. A sostegno di quanto affermato anche le metriche calcolate riportano valori molto alti con una Accuracy pari a 1 e un F-1 score anch'esso pari a 1. Analizzando i grafici della curva ROC e della curva AUC, allo stesso modo, è possibile notare come l'algoritmo K-NN ha raggiunto delle performance molto buone. Nel caso della curva ROC la curva si avvicina molto all'angolo in alto a sinistra e cioè è molto vicina al valore 1; ciò significa che è stato ottenuto un buon test.

Sebbene il modello appena citato abbia ottenuto delle performance molto buone, vi sono ancora molte ricerche da effettuare nel campo.

Nel seguente studio, per limiti di tempo, sono state applicate esclusivamente tecniche di apprendimento supervisionato. Ciò significa che gli algoritmi di apprendimento imparano a riconoscere i dati a partire da una serie di dati etichettati quindi che hanno una label.
Sarebbe auspicabile che ulteriori ricerche future sondassero l'analisi con ulteriori modelli di apprendimento automatico quali ad esempio il clustering. Il clustering può essere definito come un insieme di tecniche per l'analisi multivariata, cioè quelle parte della statistica in cui l'oggetto dell'analisi è formato da almeno due componenti, aventi l'obiettivo di selezionare e raggruppare oggetti omogenei in un dataset.

Il clustering rientra tra le tecniche apprendimento non supervisionata in cui l'algoritmo si allena basandosi sulle proprietà comuni dei dati, raggruppa cioè i dati per proprietà simili.

In secondo luogo, si potrebbe completare lo studio includendo nel confronto anche ulteriori tecniche di feature selection. Per motivi di tempo nella seguente sperimentazione sono state utilizzate le due tecniche precedentemente citate, quali: la tecnica Anova e la Mutual Information.

Infine, sarebbe interessante affrontare lo stesso studio applicato ad un dataset di dimensioni maggiori rispetto alla quantità di osservazioni disponibili, così da verificare come cambiano le prestazioni delle metodologie di feature selection qui trattate.

Acronimi

AI: Artificial Intelligence
AUC: Area Under the Roc Curve
BCG: Boston Consulting Group
CAC: Costumer Acquisition Cost
CRM: Customer Relationship Management
DT: Decision Tree
FP: False Positive
FN: False Negative
KNN: K-Nearest Neighbors
KPI: Key Performance Indicator
LDA: Linear Discriminant Analysis
ML: Machine Learning
NB: Naïve-Bayes
PCA: Principal Component Analysis
PMI: Piccole Medie Imprese
RF: Random Forest
RL: Regressione Lineare
ROC: Receiver Operating Characteristics
SVM: Support Vector Machine
TP: True Positive
TN: True Negative

Glossario figure

Appendice

In questa sezione viene fornito il codice che è stato utilizzato nella sperimentazione ed i relativi commenti esplicativi del codice.

```
# -*- coding: utf-8 -*-
"""
Created on Fri Mar 17 17:58:34 2023

@author: ADMIN
"""
```

I seguenti codici preliminari permettono di scaricare le librerie e i pacchetti necessari ai fini della sperimentazione. In questo studio sono state importate complessivamente cinque librerie con funzioni ben distinte. Queste sono rispettivamente: numpy, pandas, matplotlib, seaborn e scikit-learn. Dalla libreria di scikit-learn sono stati importati poi ulteriori sottopacchetti utili ai fini computazionali.

```
#%%
#Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns


#%%
#Sklearn modules
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.metrics import classification_report,
confusion_matrix, plot_confusion_matrix, roc_curve,
roc_auc_score
from sklearn.model_selection import train_test_split,
cross_val_score, StratifiedKFold, GridSearchCV
from sklearn.feature_selection import SelectKBest, chi2,
mutual_info_regression
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.decomposition import PCA

#Statistics
from statistics import median

#%%
```

Nella cella sottostante sono stati importati i tre dataset che sono stati denominati rispettivamente “df1”, “df2”, “df3”. E successivamente sono stati uniti con la funzione *merge* di pandas che ha restituito un unico dataframe costituito complessivamente da 40 variabili.

```
# **Reading the data**
df1 = pd.read_csv("data1.csv")

df2 = pd.read_csv("data2.csv")

df3 = pd.read_csv("data3.csv")

df4 = pd.merge(pd.merge(df1,df2,on='id'),df3, on='id')

print(df4)
```

a questo punto si è passati alla fase di esplorazione del dataset che permette di analizzare i dati e fare le opportune considerazioni prima di modificare il set di dati nella fase di preprocessing.
La funzione *info()* fornisce informazioni relative al dataframe. Le informazioni contengono il numero di colonne, il numero di etichette, il tipo di dati, la memoria utilizzata e il numero di celle in ciascuna colonna.
La funzione *sample()* restituisce un campione casuale di elementi da un asse dell’oggetto.
La funzione shape() restituisce una tupla che rappresenta la dimensionalità del dataframe e cioè fornisce una descrizione del numero di righe e del numero di colonne presenti nel dataset.
La funzione *head()* fornisce un numero specifico di righe, stringa dall’alto. Se non viene specificato un numero la funzione tendenzialmente restituisce le prime 5 righe. È utile per testare rapidamente se l’oggetto ha il giusto tipo di dati.
La funzione *describe()* invece genera statistiche descrittive. Le statistiche descrittive includono quelle che riassumono la tendenza centrale, la dispersione e la forma della distribuzione di un set di dati, esclusi i valori NaN. L’indice del risultato include: il conteggio, la media, la deviazione standard, il min., il max., il percentile inferiore, 50 e il percentile superiore. In questo caso il percentile inferiore è pari a 25, il percentile superiore è 75 mentre il 50esimo percentile corrisponde alla mediana.

```
#%%

# Is used to get a concise summary of the DataFrame including
the index dtype and columns, non-null values and memory usage
df4.info()
```

```
# Return a random sample of items from an axis of object
df4.sample(15)
#%%
# **Data Exploration**

# Returns a tuple (number of rows, number of
columns)representing the dimensionality of the DataFrame
df4.shape
#%%
# Is used to get a concise summary of the DataFrame including
the index dtype and columns, non-null values and memory usage
df4.info()

#%%
# This function returns the first n rows for the object based on
position. It is useful for quickly testing if your object has
the right type of data in it.
df4.head()
#%%

# Generate descriptive statistics.
df4.describe()

#%%

# Create arrays with descriptions so we can see all in the
variable explorer
describe = df4.describe()

#%%

# Returns description of the data in the DataFrame, now
including the object
df4.describe(include="object")
describe_object = df4.describe(include="object")
```

In questa sezione è stata effettuata la pulizia del dataset. Con la funzione *isna().sum() di pandas* è stato individuato il numero di valori mancanti per ciascuna colonna. Successivamente è stato calcolato il numero complessivo di valori mancanti all'interno del dataset con la funzione *isnull().values.sum*(). Una volta individuato il numero totale di valori mancanti, sono stati rappresentati in un grafico a barre con la funzione *sns.barplot* di matplotlib che ha permesso di raffigurare per ciascuna variabile il numero di valori mancanti espressi in percentuale.

Dopo aver effettuato questa analisi preventiva, sono stati eliminate le variabili contenenti almeno il 50% di NaN tramite la funzione *drop()* di pandas. Con la stessa funzione sono state eliminate le variabili che non concorrono a fornire informazioni utili allo studio.

```
#%%
# **Data Cleaning**

# Features' arrays
feature_list = list(df4.columns)

#%%

# Let's see all the labels that are object
df4.select_dtypes('O').info()

#%%

# Function to get count of missing values in each column
missing_values_df = df4.isna().sum()
missing_values_df

#%%

# Count the total number of missing values
print(df4.isnull().values)
print(type(df4.isnull().values))
print(df4.isnull().values.sum())  # there are 1130954 NaN
#%%
# Plot of percentage of missing values
plt.xticks(rotation='vertical')
sns.barplot(x=missing_values_df.index, y=missing_values_df)
plt.title('Percentage of missing values')
#%%

# Delete features with 50% or more missing values
df4.drop(['activity_new', 'campaign_disc_ele',
'date_first_activ', 'forecast_base_bill_ele',
'forecast_base_bill_year', 'forecast_bill_12m',
'forecast_cons'], axis=1, inplace=True)
#%%
# Delete features that we do not consider relevant
df4.drop(['id', 'date_activ', 'date_end', 'price_date'], axis=1,
inplace=True)
#%%

# The percentage of missing values on the whole dataset does not
exceed 55%, so we decide to replace the missing values with
```

```
mean, mode and median depending on the type and problems of the
variables
```

È stata poi eseguita una analisi e relativa pulizia di ciascuna variabile.
Con la funzione *value.counts()* è stato conteggiato il numero di record per ogni combinazione di valori univoci per ciascuna colonna.
Con la funzione *fillna().mode()* sono stati sostituiti i Nan delle variabili categoriche con la moda, mentre con la funzione *fillna().median()* sono stati sostituiti i Nan delle variabili numeriche utilizzando in questo caso la mediana.
La funzione *boxplot()* di matplotlib ha permesso di verificare la presenza di eventuali outliers all'interno delle variabili. Nel caso in cui fossero presenti gli outliers, questi sono stati sostituiti con i valori della mediana tramite la funzione *replace_numerical_outliers().median().* Inoltre, con la funzione *drop(< 0).index* sono stati eliminati i valori negativi che compromettono l'attendibilità delle analisi.
Infine, con la funzione *isna().sum* è stato verificato che tutti i Nan fossero stati eliminati in modo da procedere correttamente con le fasi successive della sperimentazione.

```
#%%
# Aanalisis and data cleaning of each variable

# channel_sales
channel_sales = df4['channel_sales'].value_counts(dropna=False)
#%%
# Nan replacement with channel_sales mode
df4['channel_sales'] =
df4['channel_sales'].fillna(df4['channel_sales'].mode()[0])

#%%
#cons_12m
cons_12m = df4['cons_12m'].value_counts(dropna=False)
#%%
# From the boxplot we see that there are outliers
boxplot = df4.boxplot(column=['cons_12m'],grid=False, rot=45,
fontsize=15)
#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[cons_12m].median()
    std = df4[cons_12m].std()
    outliers = ((df4[cons_12m] - median).abs()) > z_thresh*std
    df4[outliers] = np.nan
    df4[cons_12m].fillna(median, inplace=True)
    #%%
# Replace outliers with the median of cons_12m
```

```
#df4['cons_12m'] =
df4['cons_12m'].fillna(df4['cons_12m'].median())
#%%
#cons_gas_12m
cons_gas_12m = df4['cons_gas_12m'].value_counts(dropna=False)
#%%
# From the boxplot we see that there are outlaier
boxplot = df4.boxplot(column=['cons_gas_12m'],grid=False,
rot=45, fontsize=15)
#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[cons_gas_12m].median()
    std = df4[cons_gas_12m].std()
    outliers = ((df4[cons_gas_12m] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[cons_gas_12m].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['cons_gas_12m'] =
df4['cons_gas_12m'].fillna(df4['cons_gas_12m'].median())
#%%
#cons_last_month
cons_last_month=
df4['cons_last_month'].value_counts(dropna=False)

#%%
# Delete the negative number
df4.drop(df4[df4['cons_last_month'] < 0].index, axis=0,
inplace=True)
#%%
# From the boxplot we see that there are outlier
boxplot = df4.boxplot(column=['cons_last_month'],grid=False,
rot=45, fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[cons_last_month].median()
    std = df4[cons_last_month].std()
    outliers = ((df4[cons_last_month] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[cons_last_month].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['cons_last_month'] =
df4['cons_last_month'].fillna(df4['cons_last_month'].median())
#%%
#date_modif_prod
```

```
date_modif_prod =
df4['date_modif_prod'].value_counts(dropna=False)
#%%
# Nan replacement with date_modif_prod mode
df4['date_modif_prod'] =
df4['date_modif_prod'].fillna(df4['date_modif_prod'].mode()[0])

#%%
# date_renewal
date_renewal = df4['date_renewal'].value_counts(dropna=False)
#%%
# Nan replacement with date_renewal mode
df4['date_renewal'] =
df4['date_renewal'].fillna(df4['date_renewal'].mode()[0])
#%%
#forecast_cons_12m
forecast_cons_12m  =
df4['forecast_cons_12m'].value_counts(dropna=False)
#%%
# From the boxplot we see that there are outlaier
boxplot = df4.boxplot(column=['forecast_cons_12m'],grid=False,
rot=45, fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[forecast_cons_12m].median()
    std = df4[forecast_cons_12m].std()
    outliers = ((df4[forecast_cons_12m] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[forecast_cons_12m].fillna(median, inplace=True)
# Replace outliers with the median of cons_12m
#df4['forecast_cons_12m'] =
df4['forecast_cons_12m'].fillna(df4['forecast_cons_12m'].median(
))
#%%

#forecast_cons_year
forecast_cons_year =
df4['forecast_cons_year'].value_counts(dropna=False)
#%%
# From the boxplot we see that there are outlaier
boxplot = df4.boxplot(column=['forecast_cons_year'],grid=False,
rot=45, fontsize=15)
#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[forecast_cons_year].median()
    std = df4[forecast_cons_year].std()
```

```
    outliers = ((df4[forecast_cons_year] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[forecast_cons_year].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['forecast_cons_year'] =
df4['forecast_cons_year'].fillna(df4['forecast_cons_year'].media
n())
#%%

#forecast_discount_energy
forecast_discount_energy =
df4['forecast_discount_energy'].value_counts(dropna=False)
#%%
# Replace the missing values with its median
df4['forecast_discount_energy'] =
df4['forecast_discount_energy'].fillna(df4['forecast_discount_en
ergy'].median())
#%%
#forecast_meter_rent_12m
forecast_meter_rent_12m =
df4['forecast_meter_rent_12m'].value_counts(dropna=False)
#%%
# From the boxplot we see that there are outlaier
boxplot =
df4.boxplot(column=['forecast_meter_rent_12m'],grid=False,
rot=45, fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[forecast_meter_rent_12m].median()
    std = df4[forecast_meter_rent_12m].std()
    outliers = ((df4[forecast_meter_rent_12m] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[forecast_meter_rent_12m].fillna(median, inplace=True)
#%%
#%%
# Replace outliers with the median of cons_12m
#df4['forecast_meter_rent_12m'] =
df4['forecast_meter_rent_12m'].fillna(df4['forecast_meter_rent_1
2m'].median())
#%%

#forecast_price_energy_p1
forecast_price_energy_p1 =
df4['forecast_price_energy_p1'].value_counts(dropna=False)
#%%
```

```
# Replace the missing values with its median
df4['forecast_price_energy_p1'] =
df4['forecast_price_energy_p1'].fillna(df4['forecast_price_energ
y_p1'].median())
#%%
# From the boxplot we see that there are outlaier
boxplot =
df4.boxplot(column=['forecast_price_energy_p1'],grid=False,
rot=45, fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[forecast_price_energy_p1].median()
    std = df4[forecast_price_energy_p1].std()
    outliers = ((df4[forecast_price_energy_p1] - median).abs())
> z_thresh*std
    df4[outliers] = np.nan
    df4[forecast_price_energy_p1].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['forecast_price_energy_p1'] =
df4['forecast_price_energy_p1'].fillna(df4['forecast_price_energ
y_p1'].median())
#%%
#forecast_price_energy_p2
forecast_price_energy_p2 =
df4['forecast_price_energy_p2'].value_counts(dropna=False)
#%%
# Replace the missing values with its median
df4['forecast_price_energy_p2'] =
df4['forecast_price_energy_p2'].fillna(df4['forecast_price_energ
y_p2'].median())
#%%
#forecast_price_pow_p1
forecast_price_pow_p1 =
df4['forecast_price_pow_p1'].value_counts(dropna=False)
#%%
# Delete the negative number
df4.drop(df4[df4['forecast_price_pow_p1'] < 0].index, axis=0,
inplace=True)
#%%
# Replace the missing values with its median
df4['forecast_price_pow_p1'] =
df4['forecast_price_pow_p1'].fillna(df4['forecast_price_pow_p1']
.median())
#%%
#has_gas
has_gas = df4['has_gas'].value_counts(dropna=False)
#%%
```

```
#imp_cons
imp_cons = df4['imp_cons'].value_counts(dropna=False)

#%%
# From the boxplot we see that there are outlaier
boxplot = df4.boxplot(column=['imp_cons'],grid=False, rot=45,
fontsize=15)

#%%
# Replace outliers with the median of cons_12m
#df4['imp_cons'] =
df4['imp_cons'].fillna(df4['imp_cons'].median())
#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[imp_cons].median()
    std = df4[imp_cons].std()
    outliers = ((df4[imp_cons] - median).abs()) > z_thresh*std
    df4[outliers] = np.nan
    df4[imp_cons].fillna(median, inplace=True)
    #%%
#margin_gross_pow_ele
margin_gross_pow_ele =
df4['margin_gross_pow_ele'].value_counts(dropna=False)
#%%
# Delete the negative number
df4.drop(df4[df4['margin_gross_pow_ele'] < 0].index, axis=0,
inplace=True)
#%%
# Replace the missing values with its median
df4['margin_gross_pow_ele'] =
df4['margin_gross_pow_ele'].fillna(df4['margin_gross_pow_ele'].m
edian())
#%%
# From the boxplot we see that there are outlaier
boxplot =
df4.boxplot(column=['margin_gross_pow_ele'],grid=False, rot=45,
fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[margin_gross_pow_ele].median()
    std = df4[margin_gross_pow_ele].std()
    outliers = ((df4[margin_gross_pow_ele] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[margin_gross_pow_ele].fillna(margin_gross_pow_ele,
inplace=True)
    #%%
# Replace outliers with the median of cons_12m
```

```
#df4['margin_gross_pow_ele'] =
df4['margin_gross_pow_ele'].fillna(df4['margin_gross_pow_ele'].m
edian())
#%%
#margin_net_pow_ele
margin_net_pow_ele =
df4['margin_net_pow_ele'].value_counts(dropna=False)
#%%
# Delete the negative number
df4.drop(df4[df4['margin_net_pow_ele'] < 0].index, axis=0,
inplace=True)
#%%
#Replace the missing values with its median
df4['margin_net_pow_ele'] =
df4['margin_net_pow_ele'].fillna(df4['margin_net_pow_ele'].media
n())
#%%
# From the boxplot we see that there are outlaier
boxplot = df4.boxplot(column=['margin_net_pow_ele'],grid=False,
rot=45, fontsize=15)
#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[margin_net_pow_ele].median()
    std = df4[margin_net_pow_ele].std()
    outliers = ((df4[margin_net_pow_ele] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[margin_net_pow_ele].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['margin_net_pow_ele'] =
df4['margin_net_pow_ele'].fillna(df4['margin_net_pow_ele'].media
n())
#%%
#nb_prod_act
nb_prod_act = df4['nb_prod_act'].value_counts(dropna=False)
#%%
#net_margin
net_margin = df4['net_margin'].value_counts(dropna=False)
#%%
#Replace the missing values with its median
df4['net_margin'] =
df4['net_margin'].fillna(df4['net_margin'].median())
#%%
# Delete the negative number
df4.drop(df4[df4['net_margin'] < 0].index, axis=0, inplace=True)
#%%
# From the boxplot we see that there are outlaier
```

```
boxplot = df4.boxplot(column=['net_margin'],grid=False, rot=45,
fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[net_margin].median()
    std = df4[net_margin].std()
    outliers = ((df4[net_margin] - median).abs()) > z_thresh*std
    df4[outliers] = np.nan
    df4[net_margin].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['net_margin'] =
df4['net_margin'].fillna(df4['net_margin'].median())
#%%
#num_years_antig
num_years_antig =
df4['num_years_antig'].value_counts(dropna=False)
#%%
#origin_up
origin_up = df4['origin_up'].value_counts(dropna=False)
#%%
# Nan replacement with date_renewal mode
df4['origin_up'] =
df4['origin_up'].fillna(df4['origin_up'].mode()[0])
#%%
#pow_max
pow_max = df4['pow_max'].value_counts(dropna=False)
#%%
#Replace the missing values with its median
df4['pow_max'] = df4['pow_max'].fillna(df4['pow_max'].median())

#%%
# From the boxplot we see that there are outlaier
boxplot = df4.boxplot(column=['pow_max'],grid=False, rot=45,
fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[pow_max].median()
    std = df4[pow_max].std()
    outliers = ((df4[pow_max] - median).abs()) > z_thresh*std
    df4[outliers] = np.nan
    df4[pow_max].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['pow_max'] = df4['pow_max'].fillna(df4['pow_max'].median())
#%%
#price_p1_var
```

```
price_p1_var = df4['price_p1_var'].value_counts(dropna=False)
#%%
#Replace the missing values with its median
df4['price_p1_var'] =
df4['price_p1_var'].fillna(df4['price_p1_var'].median())
#%%
# There are outliers
boxplot = df4.boxplot(column=['price_p1_var'],grid=False,
rot=45, fontsize=15)

#%%
def replace_numerical_outliers(df4, column_name, z_thresh=3):
    median = df4[price_p1_var].median()
    std = df4[price_p1_var].std()
    outliers = ((df4[price_p1_var] - median).abs()) >
z_thresh*std
    df4[outliers] = np.nan
    df4[price_p1_var].fillna(median, inplace=True)
#%%
# Replace outliers with the median of cons_12m
#df4['price_p1_var'] =
df4['price_p1_var'].fillna(df4['price_p1_var'].median())
#%%
#price_p2_var
price_p2_var = df4['price_p2_var'].value_counts(dropna=False)
#%%
#Replace the missing values with its median
df4['price_p2_var'] =
df4['price_p2_var'].fillna(df4['price_p2_var'].median())
#%%
#price_p3_var
price_p3_var = df4['price_p3_var'].value_counts(dropna=False)
#%%
#Replace the missing values with its median
df4['price_p3_var'] =
df4['price_p3_var'].fillna(df4['price_p3_var'].median())
#%%
#price_p1_fix
price_p1_fix = df4['price_p1_fix'].value_counts(dropna=False)
#%%
#Replace the missing values with its median
df4['price_p1_fix'] =
df4['price_p1_fix'].fillna(df4['price_p1_fix'].median())
#%%
# Delete the negative number
df4.drop(df4[df4['price_p1_fix'] < 0].index, axis=0,
inplace=True)
#%%
#price_p2_fix
```

```
price_p2_fix = df4['price_p2_fix'].value_counts(dropna=False)
#%%
#Replace the missing values with its median
df4['price_p2_fix'] =
df4['price_p2_fix'].fillna(df4['price_p2_fix'].median())

#%%
#price_p3_fix
price_p3_fix = df4['price_p3_fix'].value_counts(dropna=False)

#%%
#Replace the missing values with its median
df4['price_p3_fix'] =
df4['price_p3_fix'].fillna(df4['price_p3_fix'].median())
#%%
#churn
churn = df4['churn'].value_counts(dropna=False)
#%%

# Check whether it has eliminated all the missing values
missing_values_df4 = df4.isna().sum()
missing_values_df4
```

La sperimentazione procede con l'Encoding delle variabili categoriche in modo da rendere intellegibile alla macchina i vari livelli delle variabili qualitative.

In un primo momento, con la funzione *select_dtypes(include=" object")* sono state selezionate esclusivamente le variabili categoriche e tramite la funzione *LabelEncoder()* di scikit-learn sono state trasformate in valori numerici.

Successivamente è stata estratta la y cioè la variabile target tramite la funzione *y = new.churn* poiché in questo caso la variabile di riferimento è "churn".

```
#%%
# Encoding categorical variables

# Select only categorical features
features_categoriche = df4.select_dtypes(include="object")

#%%
# channel_sales
channel_sales_le = LabelEncoder()
df4['channel_sales'] =
channel_sales_le.fit_transform(df4['channel_sales'])
channel_sales_le.classes_
df4['channel_sales'].value_counts(dropna=False)
```

```
#%%

# date_modif_prod

date_modif_prod = LabelEncoder()
df4['date_modif_prod'] =
date_modif_prod.fit_transform(df4['date_modif_prod'])
date_modif_prod.classes_
df4['date_modif_prod'].value_counts(dropna=False)

#%%
# date_renewal
date_renewal_le = LabelEncoder()
df4['date_renewal'] =
date_renewal_le.fit_transform(df4['date_renewal'])
date_renewal_le.classes_
df4['date_renewal'].value_counts(dropna=False)
#%%
#has_gas

df4['has_gas'].value_counts()
m = {
    "f": 0,
    "t": 1,
}
df4['has_gas'] = df4['has_gas'].map(m)
#%%
#origin_up
origin_up_le = LabelEncoder()
df4['origin_up'] = origin_up_le.fit_transform(df4['origin_up'])
origin_up_le.classes_
df4['origin_up'].value_counts(dropna=False)
#%%

# Check to see if there are no more categorical variables
df4.info()
#%%
df4_dict = df4.to_dict('records')
df4_list_bal = list()
count=0
for row in df4_dict:
    if row['churn'] == 0:
        count += 1
        if count<=17730:
            df4_list_bal.append(row)
    else:
        df4_list_bal.append(row)

df4_new = pd.DataFrame(df4_list_bal)
```

```
#%%
# Extraction of the y
y = df4_new.churn
df4_new = df4_new.drop('churn', axis=1)

#%%
# Check if y has missing values
y.isna().sum()  # no
#%%

# Check if I have an unbalanced dataset
y.value_counts()
z = y.value_counts()
xdata3 = ['0', '1']

fig2 = plt.figure(figsize=(10, 7))

plt.bar(xdata3, z)
plt.xticks(np.linspace(0, 12, 13, endpoint=True))
plt.show()

# Is somewhat unbalanced in favor of the 0(clienti che hanno
abbandonato), penalizing the 1
```

Nella sezione denominata “Statistical Analysis” sono state svolte le analisi statistiche per analizzare la correlazione tra le singole variabili e la variabile target ed eliminare le eventuali informazioni ridondanti. Più precisamente, con la funzione *corr()* è stato possibile analizzare la relazione tra le variabili e con la funzione *sns.heatmap()* è stata rappresentata in una mappa di calore la correlazione di ciascuna variabile con la variabile target.

Oltre all’analisi di correlazione è stata applicata l’analisi a componenti principali. È stata applicata l’ACP perché è una tecnica di riduzione della dimensionalità lineare che permette di estrarre informazioni da uno spazio ad alta dimensione proiettandole in uno spazio secondario di dimensione inferiore. Essa infatti cerca di preservare le parti essenziali con maggiore variazione dei dati e rimuove le parti non essenziali. Con la funzione *StandardScaler()* di scikit-learn sono stati standardizzati tutti i dati. Poi è stata definita una soglia pari al 95% della varianza spiegata per determinare il numero di componenti principali. E infine con la funzione *pca.n_components* sono state selezionate le componenti principali con maggiore varianza spiegata.

```
#%%
# **Statistical Analysis**
```

```
# Correlation of problematic variables with churn
corr = df4_new.corr()
#%%
# Correlation between features and churn
df4_new['channel_sales'].corr(y)
df4_new['cons_12m'].corr(y)
df4_new['cons_gas_12m'].corr(y)
df4_new['cons_last_month'].corr(y)
df4_new['date_modif_prod'].corr(y)
df4_new['date_renewal'].corr(y)
df4_new['forecast_cons_12m'].corr(y)
df4_new['forecast_cons_year'].corr(y)
df4_new['forecast_discount_energy'].corr(y)
df4_new['forecast_meter_rent_12m'].corr(y)
df4_new['forecast_price_energy_p1'].corr(y)
df4_new['forecast_price_energy_p2'].corr(y)
df4_new['forecast_price_pow_p1'].corr(y)
df4_new['has_gas'].corr(y)
df4_new['imp_cons'].corr(y)
df4_new['margin_gross_pow_ele'].corr(y)
df4_new['margin_net_pow_ele'].corr(y)
df4_new['nb_prod_act'].corr(y)
df4_new['net_margin'].corr(y)
df4_new['num_years_antig'].corr(y)
df4_new['origin_up'].corr(y)
df4_new['pow_max'].corr(y)
df4_new['price_p1_var'].corr(y)
df4_new['price_p2_var'].corr(y)
df4_new['price_p3_var'].corr(y)
df4_new['price_p1_fix'].corr(y)
df4_new['price_p2_fix'].corr(y)
df4_new['price_p3_fix'].corr(y)
#%%
# Correlation between features of the dataset
rounded_corr_matrix = df4_new.corr().round(2)
#%%
# Heatmap on all variables
heatmap = sns.heatmap(rounded_corr_matrix, annot=True)
heatmap.set_title('Correlation Heatmap', fontdict={'fontsize':
12}, pad=12)
#%%

# We choose few features, the central ones in the overall
heatmap, the ones that are most related
features = ["forecast_cons_12m", "forecast_cons_year",
           "forecast_discount_energy",
"forecast_meter_rent_12m", "forecast_price_energy_p1",
           "forecast_price_energy_p2", "forecast_price_pow_p1",
"net_margin",
```

```
            "num_years_antig","origin_up", "pow_max",
"price_p1_var", "price_p2_var", "price_p3_var", "price_p1_fix",
"price_p2_fix", "price_p3_fix"]

subset = rounded_corr_matrix[features].loc[features]
heatmap1 = sns.heatmap(subset, annot=True)
#%%
# Delete forecast_discount_energy,num_years_antig and origin_up
features2 = ["forecast_cons_12m", "forecast_cons_year",
             "forecast_meter_rent_12m",
"forecast_price_energy_p1",
             "forecast_price_energy_p2",
"forecast_price_pow_p1", "net_margin",
             "pow_max", "price_p1_var", "price_p2_var",
"price_p3_var", "price_p1_fix", "price_p2_fix", "price_p3_fix"]


subset2 = rounded_corr_matrix[features2].loc[features2]
heatmap2 = sns.heatmap(subset2, annot=True)
#%%

# ACP
# Standardized all values
sc = StandardScaler()
df_scaled = sc.fit_transform(df4_new)
#%%
# Pick 6 components
pca = PCA(n_components=6)

df6 = pca.fit_transform(df_scaled)

explained_variance = pca.explained_variance_ratio_
#%%
# Choice of components based on 95% variance
pca = PCA(.95)

df7 = pca.fit_transform(df_scaled)

explained_variance = pca.explained_variance_ratio_

pca.n_components_
#%%
# Try with 90% of variance
pca = PCA(.90)

df8 = pca.fit_transform(df_scaled)

explained_variance = pca.explained_variance_ratio_
```

```
pca.n_components_
```

In questa sezione vengono applicate le tecniche di *feature selection* che permettono di selezionare dal dataset le features più utili ai fini della sperimentazione e garantire dei risultati più accurati.

Le tecniche di features selection applicate in questo studio sono: la tecnica Anova con la funzione *SelectKBest(f_classif*) di scikit-learn, e la tecnica Mutual Information con la funzione *mutual_info.sort_values()* sempre di scikit-learn.

```
#%%
# **Features Selection**
# Define x
x = df4_new

#%%
#ANOVA UNIVARIATE Feature Selection
# Split dataset to select feature and evaluate the classifier
x_train, x_test, y_train, y_test = train_test_split(x, y,
stratify=y, random_state=0)

#%%
from sklearn.feature_selection import SelectKBest, f_classif

selector = SelectKBest(f_classif, k=4)
selector.fit(x_train, y_train)
scores = -np.log10(selector.pvalues_)
scores /= scores.max()
#%%
import matplotlib.pyplot as plt

x_indices = np.arange(x.shape[-1])
plt.figure(1)
plt.clf()
plt.bar(x_indices - 0.05, scores, width=0.2)
plt.title("Feature univariate score")
plt.xlabel("Feature number")
plt.ylabel(r"Univariate score ($-Log(p_{value})$)")
plt.show()
#%%
# Delete worst variables by score
df4_new.drop(['forecast_cons_12m', 'forecast_cons_year',
'forecast_price_energy_p1',
        'imp_cons', 'pow_max'], axis=1, inplace=True)
#%%
# Define x
x = df4_new
```

```
#%%
#MUTUAL INFORMATION feature selecion
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x, y,
test_size=0.2,random_state=0)

from sklearn.feature_selection import mutual_info_classif
mutual_info = mutual_info_classif(x_train, y_train)
mutual_info
#%%
mutual_info = pd.Series(mutual_info)
mutual_info.index = x_train.columns
mutual_info.sort_values(ascending=False)
#%%
mutual_info.sort_values(ascending=False).plot.bar(figsize=(20,
8))

#%%
from sklearn.feature_selection import SelectKBest
sel_five_cols = SelectKBest(mutual_info_classif, k=5)
sel_five_cols.fit(x_train, y_train)
x_train.columns[sel_five_cols.get_support()]
#%%
# Delete variables with mutual information less than 0.001
df4_new.drop(['forecast_discount_energy', 'has_gas',
'nb_prod_act',
       'price_p3_fix', 'price_p2_fix'], axis=1, inplace=True)
#%%
```

Una volta selezionate le variabili più importanti ai fini dello studio, sono stati applicati i modelli di apprendimento automatico che in questo caso sono quattro: Random Forest, Decision Tree, Naive Bayes e K-NN.

Prima ancora di allenare i modelli è stato suddiviso il dataset appliccando il metodo dell'hold out cross validation. Con la funzione *train_test_split(x, y, test_size=0.2, random_state=0)* il dataset è stato diviso per l'80% in train e il restante 20% in test.

In questa sezione, inoltre, sono state calcolate anche le metriche per stimare le performance di ciascun algoritmo. Sono state calcolate le seguenti metriche: la matrice di confusione con la funzione *cm = confusion_matrix(y_test, y_pred) sns.heatmap(cm, annot=True, fmt='d').set_title('confusion matrix ')*, il report di classificazione con la funzione *print(classification_report(y_test,y_pred))* contentente l'accuracy, la precision, recall e l'F1-score, la curva ROC con la funzione *ypredpropa = rfc.predict_proba(x_test)* e la curva AUC con la funzione *auc = metrics.roc_auc_score(y_test, y_pred)*

```
# **Machine Learning Models**

#RANDOM FOREST
x_train, x_test, y_train, y_test = train_test_split(x, y,

test_size=0.2,

random_state=0)
#%%
rfc = RandomForestClassifier(n_estimators=2,
                             max_depth=2,
                             random_state=0)
#%%
# Hyperparameters
p_grid = {"max_depth": [2, 3, 4, 5, 10, 15, 20, 25, 30]}

# Objective of validation stage is to tune / optimize the
hyperparameter
inner_cv = StratifiedKFold(n_splits=5, shuffle=True,
random_state=0)

# We perform the search of the best hyperparameters within the
validation set
rfc = GridSearchCV(estimator=rfc, param_grid=p_grid,
cv=inner_cv, verbose=1)
#%%
# Fit RandomForestClassifier
rfc.fit(x_train, y_train)
# Predict the test set labels
y_pred = rfc.predict(x_test)
#%%
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d').set_title('confusion matrix
')

print(classification_report(y_test,y_pred))
#%%
# ROC Curve
ypredpropa = rfc.predict_proba(x_test)

n_class = 2
fpr = {}
tpr = {}
thresh = {}

for i in range(n_class):
    fpr[i], tpr[i], thresh[i] = roc_curve(y_test, ypredpropa[:,
i], pos_label=i)
```

```
# Plotting ROC
plt.figure(figsize=(10, 7))
plt.plot(fpr[0], tpr[0], linestyle="--", color="orange",
label="Class 0 vs Rest")
plt.plot(fpr[1], tpr[1], linestyle="--", color="green",
label="Class 1 vs Rest")
plt.title("Multiclass ROC curve rfc", size=27,
fontweight="bold")
plt.xlabel("False Positive Rate", size=27, fontweight="bold")
plt.ylabel("True Positive rate", size=27, fontweight="bold")
plt.legend(loc="best")

#%%
# AUC
#roc_auc_score(y_test, rfc.predict_proba(x_test),
multi_class='ovr')

from sklearn import metrics
auc = metrics.roc_auc_score(y_test, y_pred)

false_positive_rate, true_positive_rate, thresolds =
metrics.roc_curve(y_test, y_pred)

plt.figure(figsize=(10, 8), dpi=100)
plt.axis('scaled')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.title("AUC & ROC Curve")
plt.plot(false_positive_rate, true_positive_rate, 'g')
plt.fill_between(false_positive_rate, true_positive_rate,
facecolor='lightgreen', alpha=0.7)
plt.text(0.95, 0.05, 'AUC = %0.4f' % auc, ha='right',
fontsize=12, weight='bold', color='blue')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.show()

#%%
# DECISION TREE
# Create training and testing samples
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=0)

dt = DecisionTreeClassifier(random_state=0)

# Hyperparameters
p_grid = {"max_depth": [2, 3, 4, 5, 10, 15, 20, 25, 30]}
```

```
# Objective of validation stage is to tune / optimize the
hyperparameter
inner_cv = StratifiedKFold(n_splits=5, shuffle=True,
random_state=0)

# We perform the search of the best hyperparameters within the
validation set
clf = GridSearchCV(estimator=dt, param_grid=p_grid, cv=inner_cv,
verbose=1)
#%%
# Train the model
clf.fit(x_train, y_train)
#%%
print('Best max_depth:',
clf.best_estimator_.get_params()['max_depth'])
#%%
# Accuracy of train
clf_score_train=clf.score(x_train , y_train)
clf_score_train
#%%
# Prediction
y_pred = clf.predict(x_test)

# Checking performance of our model with classification report
print(classification_report(y_test, y_pred))
#%%
# Plot our Confusion Matrix
plot_confusion_matrix(clf, x_test, y_test, cmap='Blues')
#%%
# ROC Curve
ypredpropa = clf.predict_proba(x_test)

n_class = 2
fpr = {}
tpr = {}
thresh = {}

for i in range(n_class):
    fpr[i], tpr[i], thresh[i] = roc_curve(y_test, ypredpropa[:,
i], pos_label=i)

# Plotting ROC
plt.figure(figsize=(10, 7))
plt.plot(fpr[0], tpr[0], linestyle="--", color="orange",
label="Class 0 vs Rest")
plt.plot(fpr[1], tpr[1], linestyle="--", color="green",
label="Class 1 vs Rest")
plt.title("Multiclass ROC curve clf", size=27,
fontweight="bold")
```

```
plt.xlabel("False Positive Rate", size=27, fontweight="bold")
plt.ylabel("True Positive rate", size=27, fontweight="bold")
plt.legend(loc="best")

plt.savefig("Multiclass ROC_pipe2", dpi=300)

#%%
from sklearn import metrics
auc = metrics.roc_auc_score(y_test, y_pred)

false_positive_rate, true_positive_rate, thresolds =
metrics.roc_curve(y_test, y_pred)

plt.figure(figsize=(10, 8), dpi=100)
plt.axis('scaled')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.title("AUC & ROC Curve")
plt.plot(false_positive_rate, true_positive_rate, 'g')
plt.fill_between(false_positive_rate, true_positive_rate,
facecolor='lightgreen', alpha=0.7)
plt.text(0.95, 0.05, 'AUC = %0.4f' % auc, ha='right',
fontsize=12, weight='bold', color='blue')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.show()


#%%
#Naive-Bayes
# Create training and testing samples
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.2, random_state=0)

#%%
from sklearn.naive_bayes import GaussianNB

# Build a Gaussian Classifier
model = GaussianNB()

# Model training
model.fit(x_train, y_train)

#%%
from sklearn.metrics import (
    accuracy_score,
    confusion_matrix,
    ConfusionMatrixDisplay,
    f1_score,
```

```
)

y_pred = model.predict(x_test)
accuray = accuracy_score(y_pred, y_test)
f1 = f1_score(y_pred, y_test, average="weighted")

print("Accuracy:", accuray)
print("F1 Score:", f1)
#%%
labels = [0,1]
cm = confusion_matrix(y_test, y_pred, labels=labels)
disp = ConfusionMatrixDisplay(confusion_matrix=cm,
display_labels=labels)
disp.plot();
#%%
# ROC Curve
ypredpropa = clf.predict_proba(x_test)

n_class = 2
fpr = {}
tpr = {}
thresh = {}

for i in range(n_class):
    fpr[i], tpr[i], thresh[i] = roc_curve(y_test, ypredpropa[:,
i], pos_label=i)

# Plotting ROC
plt.figure(figsize=(10, 7))
plt.plot(fpr[0], tpr[0], linestyle="--", color="orange",
label="Class 0 vs Rest")
plt.plot(fpr[1], tpr[1], linestyle="--", color="green",
label="Class 1 vs Rest")
plt.title("Multiclass ROC curve clf", size=27,
fontweight="bold")
plt.xlabel("False Positive Rate", size=27, fontweight="bold")
plt.ylabel("True Positive rate", size=27, fontweight="bold")
plt.legend(loc="best")

plt.savefig("Multiclass ROC_pipe2", dpi=300)
#%%
#AUC
# AUC
#roc_auc_score(y_test, rfc.predict_proba(x_test),
multi_class='ovr')

from sklearn import metrics
auc = metrics.roc_auc_score(y_test, y_pred)
```

```
false_positive_rate, true_positive_rate, thresolds =
metrics.roc_curve(y_test, y_pred)

plt.figure(figsize=(10, 8), dpi=100)
plt.axis('scaled')
plt.xlim([0, 1])
plt.ylim([0, 1])
plt.title("AUC & ROC Curve")
plt.plot(false_positive_rate, true_positive_rate, 'g')
plt.fill_between(false_positive_rate, true_positive_rate,
facecolor='lightgreen', alpha=0.7)
plt.text(0.95, 0.05, 'AUC = %0.4f' % auc, ha='right',
fontsize=12, weight='bold', color='blue')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.show()
#%%
# Splitting our dataset into training and test set (hold-out)
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.20, random_state=0)
[x_train.shape, y_train.shape, x_test.shape, y_test.shape]
#%%

# KNN
knn = KNeighborsClassifier(n_neighbors=3)

# Hyperparameters
p_grid = {"n_neighbors": [2, 3, 5, 4, 6, 7, 8, 9, 10],}

# Objective of validation stage is to tune / optimize the
hyperparameter
inner_cv = StratifiedKFold(n_splits=5, shuffle=True,
random_state=0)

# We perform the search of the best hyperparameters within the
validation set
knnc = GridSearchCV(knn, p_grid, n_jobs=-1, cv=inner_cv,
verbose=10, scoring="accuracy")

# Train the model
knnc.fit(x_train, y_train)

print('n_neighbors:',
knnc.best_estimator_.get_params()['n_neighbors'])

# Accuracy of train
knnc.score(x_train, y_train)

# Prediction
```

```
ypred = knnc.predict(x_test)

# Checking performance of our model with classification report
print(classification_report(y_test, ypred))

# Plot our Confusion matrix
plot_confusion_matrix(knnc, x_test, y_test, cmap='Blues')

# ROC Curve
ypredpropa = knnc.predict_proba(x_test)

n_class = 3
fpr = {}
tpr = {}
thresh = {}

for i in range(n_class):
    fpr[i], tpr[i], thresh[i] = roc_curve(y_test, ypredpropa[:,
i], pos_label=i)

# Plotting ROC
plt.figure(figsize=(10, 7))
plt.plot(fpr[0], tpr[0], linestyle="--", color="orange",
label="Class 0 vs Rest")
plt.plot(fpr[1], tpr[1], linestyle="--", color="green",
label="Class 1 vs Rest")
plt.plot(fpr[2], tpr[2], linestyle="--", color="blue",
label="Class 2 vs Rest")
plt.title("Multiclass ROC curve knnc", size=27,
fontweight="bold")
plt.xlabel("False Positive Rate", size=27, fontweight="bold")
plt.ylabel("True Positive rate", size=27, fontweight="bold")
plt.legend(loc="best")

# AUC
roc_auc_score(y_test, knnc.predict_proba(x_test),
multi_class='ovr')
#%%
```

Bibliografia:

- Shah, D., Rust, R. T., Parasuraman, A., Staelin, R., & Day, G. S. (2006). The path to customer centricity. *Journal of service research*, *9*(2), 113-124.
- Soltani, R., Nguyen, U. T., & An, A. (2018, July). A new approach to client onboarding using self-sovereign identity and distributed ledger. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)* (pp. 1129-1136). IEEE.
- King, G. J., Chao, X., & Duenyas, I. (2016). Dynamic customer acquisition and retention management. *Production and Operations Management*, *25*(8), 1332-1343.
- Aspinall, E., Nancarrow, C., & Stone, M. (2001). The meaning and measurement of customer retention. *Journal of Targeting, Measurement and Analysis for Marketing*, *10*, 79-87.
- Kumar, V., & Reinartz, W. (2018). *Customer relationship management*. Springer-Verlag GmbH Germany, part of Springer Nature 2006, 2012, 2018.
- Francis, B., & Ornati, M. (2012). *Customer Relationship Management. Teorie e Tecnologie*. Franco Angeli.
- Anderson, E. W., & Mittal, V. (2000). Strengthening the satisfaction-profit chain. *Journal of Service research*, *3*(2), 107-120.
- Hashmi, N., Butt, N. A., & Iqbal, M. (2013). Customer churn prediction in telecommunication a decade review and classification. *International Journal of Computer Science Issues (IJCSI)*, *10*(5), 271.
- Golshan Mohammadi, Reza Tavakkoli Moghaddam, and Mehrdad Mohammadi, "Hierarchical Neural Regression Models for Customer Churn Prediction", Journal of Engineering, Volume 2013 , 2013, Article ID 543940, 9 pages.
- Efraim Turban, Ramesh Sharda, Dursun Delen, Turban Efraim, Decision Support and Business Intelligence Systems.published by Pearson Education, 2007
- Ullah, I., Raza, B., Malik, A. K., Imran, M., Islam, S. U., & Kim, S. W. (2019). A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. *IEEE access*, *7*, 60134-60149.
- Umman, Tuğba, Şimşek, and Gürsoy, "Customer Churn analysis in telecommunication sector", Journal of the School of Business Administration, Vol: 39, No: 1, 2010, 35-49.
- Wouter Verbeke, Karel Dejaeger, David Martens, Joon Hur, and Bart Baesens, "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach", Expert Systems with Applications, Volume 218, Issue 1, 2012, Pages 211–229.
- arik Rashid, "Classification of Churn and nonChurn Customers for Telecommunication Companies" , International Journal of Biometrics and Bioinformatics (IJBB), Volume 3, Issue 5, 2008.
- Adnan Idris, Muhammad Rizwan , Asifullah Khan,"Churn prediction in telecom using Random Forest and PSO based data balancing in combination with various feature selection strategies", Journal of Computers and Electrical Engineering, 38 , 2012,1808–1819.

- Bingquan Huan, Mohand Tahar Kechadi, Brian Buckley, "Customer churn prediction in telecommunications", Expert Systems with Applications, 39, 2012, 1414–1425.
- Shin-Yuan Hung, David C. Yen, Hsiu-Yu Wang., "Applying data mining to telecom churn management", Expert System with Applications, 31, 2006, 515–524.
- Guangli Nie, Wei Rowe, Lingling Zhang, Yingjie Tian, Yong Shi, "Credit card churn forecasting by logistic regression and decision tree", Expert Systems with Applications, 38, 2011, 15273–15285.
- Afaq Alam Khan, Sanjay Jamwal, M.M.Sepehri, "Applying Data Mining to Customer Churn Prediction in an Internet Service Provider", International Journal of Computer Applications, vol. 9, issue 7, 2010, pp. 8-14
- Guangqing Li, Xiuqin Deng, "Customer Churn Prediction of China Telecom Based on Cluster
- Bingquan Huang, B. Buckley , T.-M. Kechadi, "Multi-objective feature selection by using NSGA-II for customer churn prediction in telecommunications", Expert Systems with Applications, 37, 2010, pp.3638–3646.
- B Larivière, D Van den Poel., "Investigating the role of product features in preventing customer churn, by using survival analysis and choice modeling: The case of financial services", Expert Systems with Applications, Volume 27, Issue 2, 2004, Pages 277– 285.
- Agarwal, V., Taware, S., Yadav, S. A., Gangodkar, D., Rao, A. L. N., & Srivastav, V. K. (2022, October). Customer-Churn Prediction Using Machine Learning. In *2022 2nd International Conference on Technological Advancements in Computational Sciences (ICTACS)* (pp. 893-899). IEEE.
- Zimek, A., & Filzmoser, P. (2018). There and back again: Outlier detection between statistical reasoning and data mining algorithms. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *8*(6), e1280.
- Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., & Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *Ieee Access*, *8*, 54776-54788.
- Schonlau, M., & Zou, R. Y. (2020). The random forest algorithm for statistical learning. The Stata Journal, 20(1), 3-29.
- Jijo, B. T., & Abdulazeez, A. M. (2021). Classification based on decision tree algorithm for machine learning. evaluation, 6, 7.
- Zhang, Z. (2016). Introduction to machine learning: k-nearest neighbors. Annals of translational medicine, 4(11).
- Dalianis, H., & Dalianis, H. (2018). Evaluation metrics and evaluation. Clinical text mining: secondary use of electronic patient records, 45-53.
- Narkhede, S. (2018). Understanding auc-roc curve. Towards Data Science, 26(1), 220-227.
- Bhakane, B. (2015). Effect of customer relationship management on customer satisfaction and loyalty. International Journal of Management (IJM) Volume, 6, 01-07.
- Reichheld, F. F., & Schefter, P. (2000). E-loyalty: your secret weapon on the web. Harvard business review, 78(4), 105-113.
- Peacock, P. R. (1998). Data mining in marketing: Part 1. Marketing Management, 6(4), 8.

- Hao, J., & Ho, T. K. (2019). Machine learning made easy: a review of scikit-learn package in python programming language. Journal of Educational and Behavioral Statistics, 44(3), 348-361.
- Vijaya, J., & Sivasankar, E. (2018). Improved churn prediction based on supervised and unsupervised hybrid data mining system. In Information and Communication Technology for Sustainable Development: Proceedings of ICT4SD 2016, Volume 1 (pp. 485-499). Springer Singapore.

Sitografia

- Churn rate: significato, definizione - GlossarioMarketing.it
- I KPI Marketing: quali sono, quando servono e come usarli in azienda (digital4.biz)
- Amy Gallo, The Value of Keeping the Right Customers (hbr.org)
- Len Markidan, "Why customer churn happens, and what you can do about it", https://www.groovehq.com/blog/ reduce-customer-churn
- Oracle, "2011 Customer experience impact report - Customer and brand relationship",http://www.oracle.com/us/products/applications/ cust-exp-impact-report-epss-1560493.pdf
- Customer Retention Versus Customer Acquisition (forbes.com)
- Frederick F. Reichheld and Phil Schefter, The Economics of E-Loyalty - HBS Working Knowledge - Harvard Business School
- Saravana kumar, Customer Retention Versus Customer Acquisition (forbes.com)
- Patricia Rioux, The Value Of Investing In Loyal Customers (forbes.com)
- sklearn.ensemble.RandomForestRegressor — scikit-learn 1.2.2 documentation
- What is Naïve Bayes | IBM
- K-nearest neighbor classifier (KNN) for k = 1 und k = 3 | Download Scientific Diagram (researchgate.net)
- Measuring Performance: The Confusion Matrix – Glass Box (glassboxmedicine.com)
- Alex McEachern, What Are the Top Customer Retention Strategies? (2023) (shopify.com)

Riassunto:

Il seguente elaborato tratta l'impatto delle tecnologie di *Machine Learning* sul Marketing e sulla gestione delle informazioni dei clienti. La sempre più crescente disponibilità di grandi quantità di dati sui clienti ha creato nuove opportunità e sfide per le aziende che desiderano utilizzare tali dati per ottenere un vantaggio competitivo. Oggigiorno, molte organizzazioni si sono accorte che le conoscenze contenute in questi enormi database rappresentano un enorme potenziale e che sono fondamentali per prendere le più opportune decisioni aziendali. In particolare, le conoscenze sui clienti sono fondamentali per l'area del Marketing.

L'aumento della concorrenza e delle scelte disponibili per i clienti ha posto nuove sfide per i marketers, che hanno dovuto adottare una prospettiva di gestione delle relazioni a lungo termine con i clienti. Questo approccio, noto come *customer relationship management*, richiede alle aziende di adattare i loro prodotti e servizi in base alle preferenze effettive dei clienti, anziché fare supposizioni generali. In questo contesto, gli strumenti di data mining detengono un enorme potenziale; questi, infatti, possono aiutare a scoprire conoscenze nascoste e a comprendere meglio i clienti. La gestione efficace delle relazioni con i clienti richiede una comprensione approfondita delle loro esigenze e preferenze. Pertanto, le analisi predittive e di gestione della conoscenza risultano essere particolarmente utili per il marketing.

Nello studio specifico presentato nella tesi, vengono delineate le tecniche di machine learning utilizzate per prevedere il tasso di abbandono dei clienti di un provider energetico neozelandese. Viene dimostrato come gli strumenti di data e analytics possono essere utili per analizzare il tasso di abbandono dei clienti e creare strategie di marketing efficaci per ridurre al minimo il numero di clienti che abbandonano. Nel contesto del settore energetico, il churn rate è un’importante metrica da monitorare, in quanto influisce direttamente sulla redditività dell’azienda. La capacità di prevedere il tasso di abbandono dei clienti può aiutare le aziende energetiche ad adottare misure preventive per ridurre il churn rate e migliorare la soddisfazione dei clienti. L'analisi sottolinea che per gestire e ridurre il tasso di abbandono, è necessario adottare un approccio predittivo basato sui dati, utilizzando algoritmi statistici e tecniche di machine learning per fare previsioni basate sui dati storici. Molti studi dimostrano che spesso la previsione del tasso di abbandono viene effettuata tramite problemi di classificazione; nello studio proposto viene applicato un task di classificazione binario. La classificazione ha come

obiettivo la costruzione di un modello che sia in grado di prevedere l'appartenenza di un'osservazione ad una specifica classe target, che può essere multinomiale oppure binaria, proprio come nel caso della classe target qui considerata "churn". Nell'approccio supervisionato, trattato nel corso della sperimentazione, il classificatore generalizza ed apprende la classificazione a partire da esempi per cui è esplicitato il corretto output.

Nel corso della ricerca, sono stati esaminati diversi algoritmi di machine learning, tra cui Decision Tree, Random Forest, Naive Bayes e il K-nn, per determinare quale metodo fosse il più adatto per la previsione del churn rate. Sono state anche analizzate diverse variabili e fattori che possono influenzare il churn rate, come il comportamento degli utenti, le caratteristiche del prodotto o servizio e i dati demografici.

I risultati della ricerca hanno dimostrato che i modelli di machine learning sono in grado di fornire previsioni accurate del churn rate. Sono state identificate le variabili più influenti e i pattern di comportamento dei clienti che indicano un potenziale churn. Ciò fornisce alle aziende l'opportunità di intervenire in modo tempestivo e adottare strategie personalizzate per ridurre il churn rate e migliorare la soddisfazione del cliente.

La struttura dell'elaborato è stata articolata come segue: nella fase introduttiva viene approfondito il tema del tasso di abbandono e del ruolo che esso riveste all'interno delle aziende. Successivamente viene presentata la fase di sperimentazione in cui vengono illustrate le varie fasi di modellazione da seguire per ottenere una buona previsione del churn del cliente. Essa si articola principalmente in tre sezioni; nella prima sezione denominata "Stato dell'arte" viene fornita un'analisi della letteratura relativa agli studi precedentemente svolti in questo contesto. Nella sezione "Materiale e metodi" è stato descritto il dataset oggetto della sperimentazione e gli strumenti utilizzati ai fini della computazione stessa, sono stati introdotti i modelli di classificazione costruiti e le misure di prestazione; nella sezione "Risultati e discussioni" viene fornita un'illustrazione e una discussione dei risultati ottenuti. Infine, nella sezione "Conlusioni e sviluppi futuri" vengono suggeriti i potenziali sviluppi futuri del lavoro svolto.

Come anticipato, nella fase introduttiva è stata definita la metrica oggetto di studio della tesi, ovvero, il churn rate.

Il churn rate anche detto tasso di abbandono o tasso di defezione, esprime la percentuale di clienti che ha abbandonato un servizio in un dato periodo di tempo rispetto al numero totale di clienti che ne ha usufruito nello stesso periodo *(Cocuzza D.).* Il churn rate è una metrica fondamentale per capire lo stato di salute di un'impresa e le sue prospettive future; essa, infatti, fa parte dei principali indicatori di Marketing, fondamentali per comprendere l'andamento di una attività.

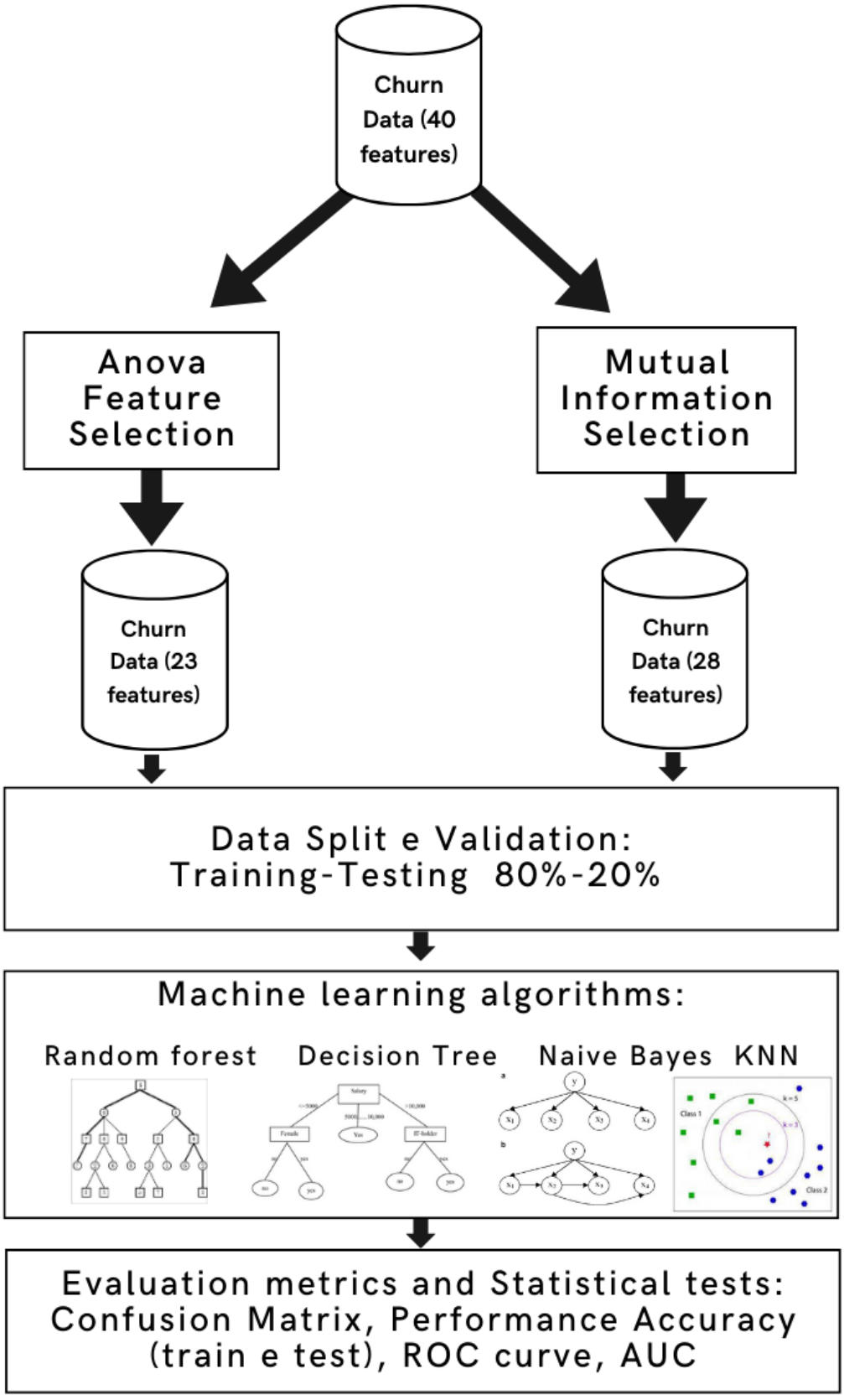
Il churn rate, quindi, è una metrica essenziale per valutare le performance di marketing del proprio business, ma non solo; questo valore può essere utilizzato anche per una serie di considerazioni aggiuntive, essenziali per curare la performance aziendale.
Ad esempio, per analizzare l'andamento del processo di fidelizzazione dei propri clienti. Può aiutare a identificare l'impatto di eventuali cambiamenti nella propria offerta, nonché a calcolare il customer lifetime value e, infine, permette di fare previsioni sulle performance future della propria attività. Comprendere il tasso di abbandono dei clienti è essenziale per valutare l'efficacia dei propri sforzi di marketing e la soddisfazione generale dei clienti.
Nella fase successiva di *analisi della letteratura* sono stati messi a confronto i vari studi precedentemente svolti sull'analisi del tasso di abbandono. Questa analisi preventiva ha permesso di contestualizzare l'argomento trattato ed analizzare le tecniche che sono state applicate in studi passati. Inoltre, è stato individuato il gap che è stato cercato di colmare nel seguente studio, ovvero: negli studi precedenti sono state condotte analisi predittive del tasso di abbandono in diversi settori ma ad oggi non sono emersi studi analoghi nel settore energetico. Con il seguente elaborato si intende smarcare questo gap e approfondire la previsione del tasso di abbandono nel settore energetico.

Successivamente è stata condotta *l'analisi sperimentale* che viene sintetizzata in cinque fasi ben distinte:

- Acquisizione dei dati: Questa fase consiste nell'acquisizione dei dati necessari per condurre l'analisi. Il set di dati per questo studio è stato acquisito da Powerco ed è stato utilizzato per analizzare le tendenze di marketing dei clienti della società neozelandese.
- Esplorazione del dataset: Lo scopo principale dell'analisi esplorativa dei dati è quello di aiutare a guardare ai dati prima di fare qualsiasi supposizione. Può aiutare a identificare

gli errori e a comprendere meglio i modelli all'interno dei dati, rilevare i valori e gli eventi anomali e a trovare interessanti relazioni tra le variabili.

- Data Pre-processing: La preelaborazione dei dati è la fase più importante nei modelli di previsione, poiché i dati sono costituiti da ambiguità, errori e ridondanze che devono essere eliminate in anticipo. I dati raccolti vengono prima aggregati e poi puliti, poiché i dati raccolti non sono adatti ai fini della modellazione.
- Estrazione delle features: In questa fase vengono identificati gli attributi per il processo di classificazione. In questo lavoro, sono state utilizzate sia variabili numeriche sia categoriche.
- Scelta del modello e metriche: In questa fase finale vengono confrontati i diversi modelli di predizione e, con il supporto delle metriche, si definisce il modello di classificazione più affidabile


Churn Data (40 features)
Anova Feature Selection
Mutual Information Selection
Churn Data (23 features)
Churn Data (28 features)
Data Split e Validation: Training-Testing 80%-20%
Machine learning algorithms:
Random forest
Decision Tree
Naive Bayes
KNN
Evaluation metrics and Statistical tests: Confusion Matrix, Performance Accuracy (train e test), ROC curve, AUC

Il lavoro di sperimentazione è stato svolto interamente su Python. Python è un linguaggio di programmazione ideato alla fine degli anni '80 e implementato nel dicembre 1989 da Guido van Rossum presso la CWI nei Paesi Bassi come successore del linguaggio ABC in grado di gestire le eccezioni e di interfacciarsi con il sistema operativo Amoeba. (Hao, J., Ho, T. K., 2019).

Il linguaggio di programmazione Python attualmente sta guadagnando un'enorme popolarità tra i data scientist e gli sviluppatori di software. A differenza del linguaggio di programmazione R, destinato principalmente all'analisi statistica dei dati, Python si presenta in una gamma molto più ampia di applicazioni, come lo sviluppo di siti Internet o applicazioni Web e desktop, l'accesso a database, il calcolo scientifico e lo sviluppo di software e giochi.

Per quanto riguarda la fase di *acquisizione dei dati*, nel seguente studio è stato utilizzato un set di dati fornito da Powerco, uno dei clienti di BCG che si occupa della fornitura di gas ed elettricità per le PMI (Piccole Medie Imprese) e per i clienti residenziali in Nuova Zelanda. L'obiettivo finale di questo studio consiste nell'individuare i clienti in declino che rischiano di cambiare fornitore, in particolare per i clienti del segmento PMI e del mercato europeo, attraverso la questione della liberalizzazione dell'energia.

Inizialmente i dati forniti da Powerco erano suddivisi in tre dataset distinti che contenevano rispettivamente:

4. Dati storici dei clienti: dati dei clienti, come l'utilizzo, la data di sottoscrizione, le previsioni di utilizzo.
5. Dati storici sui prezzi: dati storici sui prezzi, variabili e fissi.
6. Indicatore di abbandono: dati che indicano se ogni cliente ha effettuato il churning o meno.

I tre dataset sono poi stati uniti a partire dall' ID cliente, che è la variabile in comune tra tutti e tre i dataset.

Successivamente, l'*analisi esplorativa* dei dati (o EDA, exploratory data analysis) ha permesso di effettuare uno studio preventivo dei dati ottenuti e su cui è stata eseguita la sperimentazione.
Il processo di esplorazione del dataset può essere sintetizzato principalmente in due fasi:

- Screening dei dati,

- Identificazione dei valori anomali.

In primo luogo, è stato eseguito lo *screening* dei dati che permette di visualizzare i dati e comprenderne la natura.

A partire dallo screening dei dati, è stato possibile affermare che il dataset è composto da 193002 righe e 40 colonne al cui interno sono presenti complessivamente 40 variabili di natura diversa. È stato possibile poi identificare il type di ciascuna variabile; delle 40 variabili si identificano rispettivamente 22 variabili di tipo float64 e cioè numeri decimali che occupano 64 bit nella memoria del computer, 11 di tipo object che fanno riferimento a variabili categoriche e le restanti 7 di tipo int64 e cioè numeri interi. Successivamente sono stati individuati i valori mancanti all'interno del dataset che ammontano complessivamente a 1130954. Nella fase di esplorazione del dataset è fondamentale individuare questi valori per poi modificarli o eliminarli nella fase successiva in modo da non intaccare l'affidabilità delle previsioni.

Oltre ai valori mancanti, sono stati individuati i valori anomali (o outliers) per ciascuna variabile. Gli outliers, in un insieme di osservazioni, sono valori anomali e aberranti, ossia valori chiaramente distanti dalle altre osservazioni disponibili (Zimek, A., Filzmoser, 2018).

Esistono diverse tecniche per esaminare la presenza di outlier; in questo studio è stato adottato un processo che combina più tecniche, combinando tecniche di visualizzazione dei dati per esaminare la distribuzione e verificare la presenza di outlier e metodi statistici.

In primo luogo sono state analizzate le statistiche descrittive con il comando .describe() che ci ha permesso di generare alcune statistiche di riepilogo. Successivamente, dopo aver analizzato le statistiche descrittive, è stato applicato uno dei principali strumenti di visualizzazione per individuare gli outlier nei dati: i box plot.

Questo approccio è stato adottato per analizzare la presenza di eventuali outlier in ciascuna variabile e nel complesso è emerso che le variabili del dataset che presentano outlier sono tredici e sono rispettivamente: "cons_12m", "cons_gas_12m", "cons_last_month", "forecast_cons_12m", "forecast_cons_year", "forecast_meter_rent", "forecast_price_energy_p1", "imp_cons", "margin_gross_pow_ele", "margin_net_pow_ele", "net_margin", "pow_max", "price_p1_var".

Una volta analizzate le variabili all'interno del dataset e individuate le anomalie da correggere è stata condotta la *fase di preelaborazione dei dati*. La fase di preelaborazione si riferisce

all’insieme di trasformazioni applicate ai dati prima di darli in pasto all'algoritmo. La pulizia dei dati è il processo di rilevamento e correzione (o rimozione) di valori corrotti o imprecisi da un set di dati. Più precisamente si riferisce all'identificazione di parti incomplete, errate, imprecise o irrilevanti dei dati e quindi alla sostituzione, modifica o l'eliminazione dei dati sporchi o grossolani.

Indipendentemente dalla loro origine, i valori mancanti possono introdurre pregiudizi nell'analisi, ridurre la potenza dei modelli di apprendimento automatico e in generale influire sull’ accuratezza delle previsioni. Per questi motivi, la gestione dei valori mancanti è un passaggio fondamentale in questa fase. In questo caso specifico sono stati eliminati i dati non rilevanti ai fini dello studio quali, ad esempio, l’id del cliente e la data di rinnovo del contratto in quanto non concorrono a determinare il tasso di abbandono dei clienti del provider energetico. Oltre ai dati non necessari, sono state eliminate le variabili che detengono più del 50% di valori nulli che, come possiamo vedere dal grafico a barre, sono: “activity_new”, “campaign_disc_ele”, “date_first_activ”, “forecast_base_bill_ele”, “forecast_base_bill_year”, “forecast_bill_12m,” forecast_cons”. Successivamente è stata effettuata la codifica delle variabili categoriche. Per far sì che un modello operi correttamente è necessario che i dati di natura qualitativa vengano convertiti in forma numerica; per questo motivo, prima di mandare i dati in pasto al modello, è necessario applicare le opportune tecniche di encoding. Con il comando “features_categoriche = df4.select_dtypes(include="object")” sono state individuate e selezionate esclusivamente le variabili categoriche che complessivamente sono cinque: “channel_sales”, “date_modif_prod”, “date_renewal”, “has_gas”, “origin_up”.

Una volta terminato il processo di encoding è stata estratta la y che nel nostro caso è la variabile “churn”. Estraendo la variabile di interesse è possibile fare le opportune considerazioni e prevedere il tasso di abbandono dei clienti del provider.
Una volta estratta la variabile dal dataset, la stessa è stata rimossa dal dataset di appartenenza con il comando .drop('churn', axis=1). A questo punto il dataset è composto da tutte le variabili ripulite e trasformate pronte per essere mandate in pasto al modello meno la variabile churn su cui si intende fare le opportune previsioni.

Analizzato il dataset ed eseguite le opportune modifiche sulle variabili, sono state condotte le analisi statistiche. In questa fase le analisi statistiche permettono di ridurre il volume dei dati pur mantenendo la qualità dell’analisi.

In questo studio, in primo luogo, è stata condotta l'analisi di correlazione per individuare le caratteristiche che sono altamente correlate con la variabile target (churn) e che contribuiscono maggiormente alla varianza del set di dati. Per facilitare la visibilità della correlazione tra le variabili e tra queste ultime e la variabile target è stata utilizzata la mappa di calore.
Infine è stata applicata l' Analisi a Componenti Principali, anche detta PCA.
La PCA è una procedura statistica che utilizza una trasformazione ortogonale. Essa converte un gruppo di variabili correlate in un gruppo di variabili non correlate e può essere utilizzata per esaminare le relazioni tra un gruppo di variabili. Quindi, proprio come in questo studio, può essere utilizzata per la riduzione della dimensionalità. Al fine di preservare un valore alto di varianza, è stato stabilito un valore fisso pari al 95% e il modello ha restituito 16 componenti principali. Dalle 28 componenti iniziali il dataset è stato ridotto quindi a 16 componenti mantenendo, comunque, alto il livello di informazione fornito dai dati.

La *fase di selezione delle caratteristiche* è un processo molto importante dell'analisi nel data mining, poiché è la fase in cui vengono determinate le caratteristiche critiche. La selezione delle caratteristiche non solo rimuove quelle indesiderate, ma ci aiuta anche a trovare quelle più rilevanti che permettono di aumentare le prestazioni del nostro modello. Le diverse tecniche ad oggi disponibili si differenziano nel modo sia in cui cercano il sottoinsieme più caratteristico, sia in cui incorporano il modello di classificazione *(Saeys, et al., 2007).* Nello studio svolto sono state confrontate due tecniche di selezione delle caratteristiche univariate; le tecniche univariate esaminano ogni caratteristica singolarmente per determinare la forza della relazione della caratteristica con la variabile di risposta.

Esse sono rispettivamente:

3. Tecnica Anova,
4. Tecnica di Mutua informazione.

Infine, nella *fase di scelta e del modello e analisi delle metriche* sono stati allenati gli algoritmi per risolvere il task di classificazione, sono state analizzate le metriche per valutare le performance dei diversi modelli allenati e infine è stato scelto il modello che, per questo tipo di task, risulta essere più performante.

Gli algoritmi di classificazione nell'apprendimento automatico sono numerosi, e in questo lavoro, ci si è concentrati principalmente sul Random Forest, il Decision Tree, il Knn e il Naive Bayes. Per valutare le performance di questi algoritmi sono state analizzate diverse metriche, quali:

- La matrice di confusione
- Accuracy
- Precision
- Recall
- F1- score
- Curva ROC
- Curva AUC

La scelta di confrontare più metriche nello studio ha permesso di aumentare l'attendibilità dell'analisi svolta.

Dal confronto dei diversi algoritmi di apprendimento è stato possibile affermare che il modello che si presta meglio per questo tipo di task è il K-Nearest Neighbors. È possibile affermare ciò a partire da un accurato confronto dei vari modelli di classificazione. Dal confronto e dall'analisi delle metriche, infatti, è emerso che il K - NN è il modello che ha ottenuto i valori più alti e che quindi ha performato meglio con entrambe le tecniche di features selection; rispettivamente con la tecnica Anova e la Mutual Information.
Dalla matrice di confusione è stato possibile vedere come il modello abbia classificato correttamente tutte le istanze. All'interno della stessa matrice non è stato individuato nessun errore. Per questo motivo si può affermare che il classificatore K-NN si presta particolarmente bene per questo tipo di task. A sostegno di quanto affermato anche le metriche calcolate hanno riportato valori molto alti con una Accuracy pari a 1 e un F-1 score anch'esso pari a 1. Analizzando i grafici della curva ROC e della curva AUC, allo stesso modo, è stato possibile notare come l'algoritmo K-NN ha raggiunto delle performance molto buone. Nel caso della curva ROC la curva si avvicina molto all'angolo in alto a sinistra e cioè è molto vicina al valore 1; ciò significa che è stato ottenuto un buon test.

Sebbene il modello appena citato abbia ottenuto delle performance molto buone, nell'ultima sezione sono state proposte alcune ricerche da implementare in futuro.

Per limiti di tempo, infatti, sono state applicate esclusivamente tecniche di apprendimento supervisionato.
Sarebbe auspicabile che ulteriori ricerche future sondassero l'analisi con ulteriori modelli di apprendimento automatico quali ad esempio il clustering.
Il clustering rientra tra le tecniche apprendimento non supervisionata in cui l'algoritmo si allena basandosi sulle proprietà comuni dei dati, raggruppa cioè i dati per proprietà simili.
In secondo luogo, si potrebbe completare lo studio includendo nel confronto anche ulteriori tecniche di feature selection. Per motivi di tempo nella sperimentazione proposta sono state utilizzate le due tecniche precedentemente citate, quali: la tecnica Anova e la Mutual Information.
Infine, sarebbe interessante affrontare lo stesso studio applicato ad un dataset di dimensioni maggiori rispetto alla quantità di osservazioni disponibili, così da verificare come cambiano le prestazioni delle metodologie di feature selection qui trattate.

In conclusione, seppur vi siano ulteriori ricerche da effettuare in questo ambito, la tesi dimostra che l'uso di tecniche predittive di machine learning può essere uno strumento efficace per analizzare e prevedere il churn rate. Questa conoscenza può consentire alle aziende e in particolare al provider energetico di adottare misure preventive mirate e strategie di retention dei clienti, migliorando la sua redditività e competitività nel mercato.