UNIVERSITÀ
POLITECNICA
DELLE MARCHE

APPROCCI DI MACHINE LEARNING PER LA CLASSIFICAZIONE DEI DIFETTI IN AMBITO FASHION

MACHINE LEARNING APPROACHES FOR DEFECT CLASSIFICATION IN FASHION

TESI DI LAUREA TRIENNALE IN
INGEGNERIA GESTIONALE

RELATORE
PROF. ZINGARETTI PRIMO

CORRELATORE
PAOLANTI MARINA

STUDENTE
CRESCENZI CARLO

ANNO ACCADEMICO 2021/2022

Sommario

ABSTRACT

In questo elaborato viene proposto un contesto di *defect classification.* Un'azienda di capi di alta moda ha riscontrato numerosi reclami di borse dovuti a difetti di fabbrica. La gestione dei resi e dei reclami da parte di un produttore comporta ingenti danni sia patrimoniali che non patrimoniali. I danni economici possono suddividersi in spese e mancati guadagni. Le spese riguardano i costi di gestione dei reclami, quali il trasporto, lo stoccaggio dell'invenduto, costi per il trattamento dei resi che, in alcuni casi, implica anche un processo di rimessa a nuovo del prodotto difettoso. Le ricadute economiche non classificabili come costi si traducono in un mancato guadagno dovuto alla sfiducia riposta dal cliente nel marchio in seguito all'acquisto di un prodotto difettoso. Il danno all'immagine aziendale consiste quindi nella diminuzione della considerazione che i consumatori e gli investitori finanziari hanno di un'azienda.

L'intenzione delle aziende è quella di tutelarsi a fronte di questa problematica adottando soprattutto delle politiche di prevenzione dei guasti e dei difetti. La prevenzione dei resi spazia dal controllo qualità fino alla risoluzione dei guasti riscontrati dopo la consegna della merce. Nel caso in esame si vuole operare una *defect classification,* ovvero una classificazione delle entità dei difetti degli articoli reclamati dai negozi e dai clienti. L'obiettivo di questo problema di classificazione è quello di estrapolare delle correlazioni tra le entità dei difetti stessi e le caratteristiche fisiche del prodotto, per consentire in futuro all'azienda produttrice di predire la probabilità di un prodotto, che presenta determinate caratteristiche, di incorrere nel difetto durante la sua produzione. Nel corso del progetto è stato effettuato uno studio dello stato dell'arte che, oltre ad approfondire i concetti di

Artificial Intelligence e Machine Learning, ha evidenziato come l'interazione tra il mondo del fashion e quello dell'Artificial Intelligence può essere sfruttata in molteplici circostanze: ad esempio aziende che adottano delle reti neurali che analizzano delle immagini reperite dal Web per riuscire e fornire ai clienti delle *fashion recommendation*, oppure di algoritmi di Machine Learning capaci di prevedere le richieste dei clienti rendendo più efficace lo stoccaggio dei prodotto in magazzino e strumenti quali il *deep learning,* la realtà aumentata e le prove virtuali che migliorano l'esperienza di acquisto online dei clienti.

Il contributo dell'azienda al progetto in esame è stato apportato fornendo il dataset dei reclami di borse contenente inizialmente più di duemila articoli nel formato CSV (*Comma Separated Values)*. Al suo interno sono state elencate delle colonne opportunatamente etichettate dall'azienda che descrivono le caratteristiche fisiche di ciascun prodotto come il colore, i materiali da cui è composto e il modello, riferimenti alla vendita e al reclamo, come la data in cui è stato effettuato, dettagli sul destinatario, sulla data di acquisto e delle descrizioni dettagliate del difetto. I difetti sono stati valutati dal produttore secondo tre diverse entità: "ALTO", "MEDIO", "BASSO". Sarà proprio quest'ultima caratteristica, la cosiddetta *Target Label*, quella che i modelli di Machine Learning dovranno essere in grado di classificare in ciascun articolo basandosi sulle restanti features.

Trattandosi di un problema di Data Science, di conseguenza è stato adottato un metodo risolutivo denominato *CRISP-DM model* che si divide in sei fasi, ciascuna delle quali descrive le modalità di approccio alla progettazione di un problema di questo tipo.

La prima fase viene denominata *business understanding*, dove si analizza il contesto e vengono presentati gli obiettivi che si intendono conseguire. Seguono le fasi di *data undestanding* e di *data preparation*: rispettivamente, si analizza il dataset messo a disposizione dall'azienda produttrice, e si procede alla pre-elaborazione dei dati. In questa fase lo scopo è quello di ripulire il dataset da dati ridondanti, inconsistenti, rumorosi e incompleti. A tal proposito sono state eliminate delle colonne che accorpavano dei codici identificativi che erano già definiti in altre colonne, perciò non aggiungevano nessun tipo di informazione. In altri casi però è stato necessario aggiungere delle colonne tramite delle formule matematiche sviluppate in Excel, poiché considerate utili ai fini della progettazione. Infine alcune osservazioni sono state eliminate perché contenevano numerosi campi vuoti e non apportavano nessuna informazione. La fase di pre-processing è terminata solo in seguito all'applicazione di un apposito filtro alle colonne del dataset in esame: gli algoritmi di Machine Learning prediligono l'elaborazione di dati numerici piuttosto che nominali, perciò alcune features hanno subito una conversione utilizzando un *NominalToBinary*, un filtro messo a disposizione dal software *Weka*.

Weka, l'ambiente di lavoro open-source che è stato utilizzato oltre a Excel, presenta un'estesa collezione di algoritmi sia per affrontare *classification problems* come quello in esame, sia *regression problems*. L'importazione del dataset nel software è stata possibile effettuando preliminarmente una conversione del file dal formato CSV al formato ARFF, utilizzato esclusivamente per i dataset elaborati in Weka. Il dataset ultimato e pronto per essere applicato a un modello decisionale conta poco meno di un migliaio di osservazioni e ventisei features, il che lo rendono un dataset dalle dimensioni ridotte rispetto alle consuete basi di dati

applicate agli algoritmi di Machine Learning in questione. Terminata la *data preparation*, come suggerisce il CRISP-DM model, si prosegue con la fase di *modelling* dove si andrà a costruire e ad adattare il modello ottimale ai dati pre-elaborati applicando cinque algoritmi di Supervised Learning che differiscono per complessità strutturale crescente: Zero R, Decision Stump, Hoeffding Tree, Random Tree e Random Forest.

Per poter valutare in maniera adeguata i risultati dei test effettuati sul dataset da ciascun algoritmo (eccetto lo *Zero R* che funge da *benchmark*), Weka mette a disposizione alcuni indici tra cui il K-Statistic, che è un indice di accuratezza che confronta *l'observed accuracy*, ovvero il numero di istanze classificate correttamente, con *l'expected accuracy*, che indica di quanto il numero di istanze classificate si avvicina al valore reale. È stato riscontrato come il valore più alto di K-Statistic, e quindi il migliore, sia stato ottenuto in corrispondenza del Random Forest, proprio perché rientra tra gli algoritmi di *ensamble learning*, ovvero gli algoritmi che utilizzano un insieme di algoritmi per ottenere predizioni più accurate minimizzando quindi *l'overfitting*, il rischio di adattarsi troppo fedelmente ai *labeled data*.

Oltre al K-Statistic, per un corretto confronto tra i vari modelli, vengono esaminati anche il Mean Absolute Error (MAE), il Root Mean Squared Error (RMSE) e il Relative Absolute Error (RAE), i cui valori migliori sono stati riscontrati nel caso del Decision Stump.

Proseguendo con l'analisi dei risultati emerge la correlazione che persiste tra la dimensione del dataset a disposizione e la complessità dell'algoritmo applicato:

modelli semplicistici si adattano meglio a dataset di dimensioni ridotte, mentre modelli dalla complessa struttura come il Random Forest sfruttano le proprie potenzialità se applicati a dataset di dimensioni maggiori.

Un indice che fornisce un'idea di quanto il classificatore sia affine al dataset a cui viene applicato è il Relative Absolute Error, dove per affinità si intende la propensione del modello a sfruttare le proprie potenzialità e caratteristiche in relazione alla base di dati con cui interagisce. I valori ottenuti dai vari algoritmi confermano che il Random Forest, un modello molto più elaborato e complesso, è caratterizzato da un RAE di gran lunga peggiore rispetto agli altri classificatori, proprio perché la complessità del modello è sproporzionata a quella del dataset in esame, al contrario del Decision Stump.

Nonostante l'applicazione del modello del Decision Stump abbia riscontrato degli esiti complessivamente più soddisfacenti rispetto al Random Forest, quest'ultimo viene considerato il modello più affidabile grazie alla sua duttilità: il Random Forest garantisce elevate prestazioni soprattutto con dataset di dimensioni maggiori, descritti da più features e popolati da più osservazioni. L'ampliamento del dataset potrebbe significare che l'azienda, prima di avere a disposizione una base di dati adeguata allo sviluppo di una ricerca attendibile, debba incorrere in nuovi reclami continuando a subire danni economici, ma a discapito di questa supposizione si consiglia di gestire in modo differente le varie features nominali utilizzando preferibilmente un valore alfanumerico piuttosto che un valore nominale poiché facilita l'inserimento da parte dell'operatore riducendo il rischio di incorrere in dati sporchi, semplifica l'elaborazione da parte dei modelli e rende il dataset più leggibile.

In conclusione si enfatizza l'importanza della fase di *data understanding* e di reperimento dei dati, poiché disporre di un dataset più corposo aumenta sicuramente la probabilità di ottenere risultati ancora più affidabili, che in un business context come quello preso in esame, contribuiscono a ridurre i costi dei produttori nella gestione dei reclami sfruttando le potenzialità dell'Artificial Intelligence e del Machine Learning.

1 AMBITO DELLA TESI

1.1 BUSINESS CONTEXT

Un noto produttore di capi di alta moda deve far fronte alla problematica legata ai reclami e ai resi di borse difettose da parte dei negozi e dei clienti, la quale comporta ingenti costi e danni sia di carattere patrimoniale sia di carattere non patrimoniale poiché va ad intaccare l'immagine dell'azienda stessa [12]. Il danno patrimoniale consiste in una moltitudine di spese e mancati guadagni che possono riassumersi in:

- Costo della prevenzione dei resi, a sua volta divisibile in:
  - Costi di prevenzione, che prevengono l'immissione nel mercato di prodotti di bassa qualità che potrebbero dar luogo a resi;
  - Costi di valutazione, che garantiscono che il prodotto sia in linea con lo standard qualitativo che l'impresa si era prefissata;
  - Costi per risoluzione di guasti e rotture del prodotto nella fase di controllo qualità prima della spedizione della merce;
  - Costi per la risoluzione di guasti e rotture che si verificano sul prodotto dopo la consegna della merce.
- Costo di trasporto dei resi: si tratta del costo relativo al ritiro della merce presso il cliente e al trasporto fino al magazzino;
- Costo dell'Handling e immagazzinaggio dei resi: si tratta del costo relativo allo scarico della merce e alla movimentazione della merce per lo stoccaggio momentaneo nell'area prescelta del magazzino;

- Costo di trattamento dei resi: tale voce comprende sia i costi relativi alla fase di ispezione, verifica ed eventuale disassemblaggio, sia i costi relativi alle operazioni per la rimessa a nuovo;
- Costo dell'eventuale sostituzione o dell'accredito: consiste nel costo da perdita del fatturato.

Per introdurre il danno di immagine invece è opportuno precisare il concetto di reputazione aziendale, ovvero la considerazione che i consumatori e gli investitori finanziari hanno di un'azienda. Il danno all'immagine aziendale consiste nella diminuzione di questa considerazione da parte dei consumatori e delle altre società con cui l'azienda interagisce.

1.2 PROBLEM DESCRIPTION

Nell'elaborato viene descritto un contesto di *defect classification*, in cui l'azienda committente necessita di classificare le entità dei difetti dei prodotti che hanno subito un reclamo da parte degli stock o da parte dei clienti. La soluzione prevede l'applicazione di tecniche di Machine Learning che possano correttamente classificare l'entità del difetto di ciascun articolo e individuare delle correlazioni tra le caratteristiche del prodotto e l'entità del difetto. Lo scopo sarà quello di consentire al produttore di poter quantificare in futuro il rischio di incorrere nel difetto di un nuovo prodotto da immettere nel mercato basandosi solamente sulle presunte caratteristiche del prodotto stesso. Ciò implica dei costi di prevenzione, ma tutela il produttore dalle spese relative ai costi di gestione dei reclami già citati. Il lavoro svolto oggetto della tesi, offre la possibilità di interfacciarsi a dei concetti del tutto innovativi e all'avanguardia come quelli dell'Artificial Intelligence e del

Machine Learning che al giorno d'oggi costituiscono un'importante risorsa in ogni settore lavorativo e non solo.

1.3 ARTIFICIAL INTELLIGENCE E MACHINE LEARNING

Oggi Artificial Intelligence e Machine Learning vengono usati come sinonimi. In realtà l'Intelligenza Artificiale (AI) indica un ambito di ricerca indirizzato a realizzare sistemi informatici intelligenti che possano simulare la capacità di pensiero dell'uomo. Il Machine Learning è una branca dell'Artificial Intelligence che comprende una serie di tecniche in grado di gestire e analizzare in modo efficiente grandi quantità di dati, per fornire previsioni accurate, decisioni automatizzate e offrire vantaggi commerciali senza precedenti.

Le tecniche di Machine Learning attuali stanno trovando finalmente terreno fertile anche perché la potenza computazionale distribuita è oggi adeguata all'uso di queste metodologie con tempi e costi sempre più ragionevoli, per soddisfare i processi di business e le esigenze in molteplici applicazioni pratiche [4].

E' interessante notare come l'universo dell'Intelligenza Artificiale, e di conseguenza del Machine Learning, si possa dividere in due insiemi che denominano obiettivi differenti: il primo è quello della cosiddetta *Generalized AI* (detta anche AGI, *Artificial General Intelligence*), ovvero la volontà di adottare l'AI per simulare comportamenti cognitivi umani generici, il secondo è spesso connotato come *Narrow AI* (o ANI *Artificial Narrow Intelligence*) nel quale utilizziamo tecniche ottimali in contesti specifici ben definiti [8].

Per comprendere l'importanza e il reale beneficio delle soluzioni di Intelligenza Artificiale, in particolare delle applicazioni di Machine Learning, occorre partire da un concetto fondamentale: i dati. Uno dei fattori chiave per l'addestramento delle macchine e le nuove frontiere dell'Intelligenza Artificiale è rappresentato proprio dalla crescente disponibilità di dati: "*Data is the new oil*", famosa citazione del matematico Clive Humby del 2006 [8]. Negli ultimi due decenni si è assistito ad un aumento esponenziale nella quantità dell'informazione e dati che è stata immagazzinata in formato elettronico. Inoltre sono nate anche nuove posizioni lavorative inerenti alla scienza del dato, chiamate appunto *Data Scientist*. Questo fatto si deve soprattutto:

• All'incremento del potere e della velocità di calcolo degli attuali computer;

• Alla possibilità che essi hanno acquisito di avere supporti molto grandi per immagazzinare e memorizzare anche notevoli quantità di dati;

• All'introduzione di nuove tecniche che si affiancano ai tradizionali metodi di analisi statistica e che permettono di estrarre conoscenza, cioè informazioni significative (di valore), in seguito all'esplorazione di questi enormi volumi di dati [4].

L'insieme di queste tecniche prende il nome di *Data Mining*. Le applicazioni legate al mondo del data science sono innumerevoli. Partendo dal concetto più generale di Data Science, il Machine Learning può essere considerato un suo sottoinsieme più specifico, ovvero l'insieme delle metodologie utilizzate per effettuare analisi cercando di identificare correlazioni, pattern, similitudini e analogie all'interno di

gruppi di dati più o meno complessi (*Dataset*), per poter effettuare previsioni e classificazioni di varia natura. In altri termini, l'obiettivo principale del ML è quello di estrarre delle caratteristiche (si chiamano proprio *features* in terminologia ML) il più possibile distintive e rappresentative all'interno dei dati relativi ad un fenomeno che desideriamo analizzare attraverso modelli (o algoritmi di varia natura) per apprendere i meccanismi ed effettuare delle predizioni massimizzandone l'accuratezza. In sintesi il Machine Learning è un insieme di algoritmi compresi nella sfera del Data Science [8].

2 STATO DELL'ARTE

2.1 STUDIO BIBLIOGRAFICO

Lo studio dello stato dell'arte, effettuato esaminando alcuni articoli scientifici, ha confermato che non è una novità applicare l'Artificial Intelligence al mondo del fashion e spesso il primo è posto al servizio del secondo.

[Chakraborty, Samit, Md Saiful Hoque, and S. M. Surid. "A comprehensive review on image based style prediction and online fashion recommendation." Journal of Modern Technology and Engineering 5.3S (2020): 212-233]

L'Artificial Intelligence e il Machine Learning offrono numerose strumentazioni e metodi per interagire ed elaborare immagini di ogni tipologia. In questo caso il punto d'incontro tra l'ambito fashion e le tecniche di Intelligenza Artificiale è l'analisi di immagini reperite dal web. Attualmente gli studiosi distinguono due categorie di social network: gli *image-based* e i *text-based*. Naturalmente i consumatori che interagiscono col web per raccogliere informazioni che

riguardano abbigliamento accessori ed altri capi, preferiscono farlo in un *image-based social network*. Questa tipologia infatti favorisce l'accumulo di informazioni di persone di tutto il mondo, più di quanto lo facciano i social *text-based*. Gli appassionati di moda prediligono le immagini, le quali nascondono dati inerenti alla personalità dei soggetti ritratti che indossano i capi. Le aziende sfruttano la miniera di dati messa a disposizione dai social network per elaborarli attraverso reti neurali e varie tipologie di algoritmi di Machine Learning per estrapolarne informazioni utili a suggerire consigli su come vestirsi, ma soprattutto fare *style prediction* e ridurre la distanza che persiste tra la domanda del consumatore e l'offerta del produttore. L'articolo mira quindi ad evidenziare come vengono sfruttate le reti neurali e come vengono applicate le tecniche di *deep learning* per l'analisi delle immagini. L'analisi predittiva delle immagini riveste quindi un ruolo fondamentale nell'economia di una azienda di moda e testimonia come l'Artificial Intelligence sia spesso al servizio del fashion [3].

[L'AI a supporto della costumer experience durante gli acquisti online: Pillarisetty, Radhika, and Pratika Mishra. "A Review of AI (Artificial Intelligence) Tools and Customer Experience in Online Fashion Retail." International Journal of E-Business Research (IJEBR) 18.2 (2022): 1-12]

Gli autori di questa trattazione ci tengono a precisare che la considerazione che si ha comunemente dell'Artificial Intelligence è erroneamente limitata all'ambito della robotica. Questo luogo comune è destinato ad estinguersi, lasciando spazio alle evidenti potenzialità che offre l'Artificial Intelligence negli ambiti più diversificati: uno di questi è proprio il mondo del fashion, il quale ne usufruisce per migliorare sia l'esperienza dei consumatori durante la fase di acquisto online, sia la

produttività delle aziende stesse. I dati mettono in risalto come con l'avvento di Internet lo shopping online renda alle aziende più di quanto faccia la vendita nei negozi fisici, soprattutto perché l'esperienza online contribuisce notevolmente a fidelizzare il cliente nei confronti del marchio. L'e-commerce è agevolato dall'impatto avuto da strumenti quali il *deep learning*, la realtà aumentata, le prove virtuali e gli avatar che personalizzano l'esperienza di acquisto online del cliente rendendola confortevole senza tralasciare l'efficacia. L'obiettivo di questo elaborato è delineare le diverse tecnologie basate sull'Intelligenza Artificiale che influiscono sulla vendita al dettaglio online [11].

["Fashion Trend Forecasting Using Machine Learning Techniques": come delle tecniche di machine learning possono fare previsioni sui fashion trends. Chang, Audrey Aurelia, et al. "Fashion Trend Forecasting Using Machine Learning Techniques: A Review." Proceedings of the Computational Methods in Systems and Software (2021): 34-44]

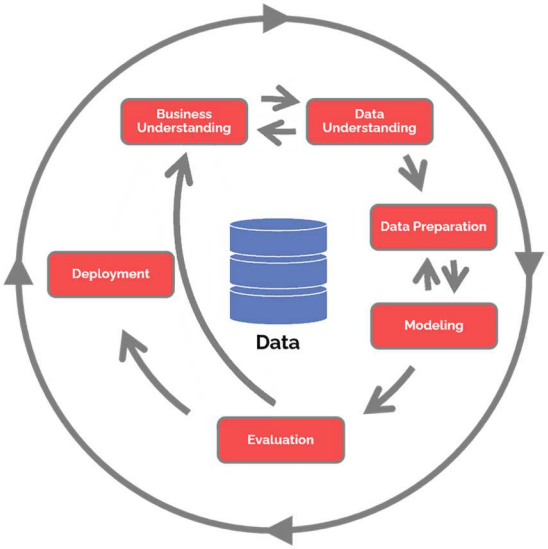
L'elaborato tratta dei vantaggi che comporta l'applicazione di tecniche di Machine Learning durante la fase di stoccaggio e produzione. Il motivo per cui riveste grande importanza poter prevedere in breve tempo i nuovi trends e le prossime richieste d'acquisto da parte dei clienti è giustificato dalla frenesia che contraddistingue la moda al giorno d'oggi. I capi d'abbigliamento possono passare in breve tempo dall'essere di tendenza a invenduti. Per questo motivo poter prevedere l'alternarsi delle tendenze nell'ambito fashion garantisce ai produttori una marcia in più, un vantaggio da sfruttare sia riducendo i costi di magazzino che quelli di smaltimento dei prodotti invenduti. L'elaborazione delle informazioni sulle preferenze dei consumatori è un processo che richiede molto lavoro, ma

rappresenta il modo più efficiente per prevedere le richieste dei consumatori, studiando inoltre l'impatto dei nuovi prodotti sul mercato. Una volta esplicati gli obiettivi e i motivi che spingono ad un tale approccio all'Artificial Intelligence in un contesto simile, vengono elencate le metodologie e gli strumenti applicati. Naturalmente comprende modelli di Machine Learning in grado di elaborare i dataset messi a disposizione ricchi di informazioni sui consumatori, sulle loro preferenze e sui loro acquisti. Fare previsioni di vendita in questa industria sta diventando sempre più impegnativo, ma al giorno d'oggi può essere considerata la chiave del successo [2].

2.2 MODELLO CRISP-DM

L'approccio alla progettazione di un problema di Data Science viene solitamente affidato al processo CRISP-DM (*CRoss-Industry Standard Process for Data Mining*) il quale si divide in sei fasi [6,8]:

Figura 1. *Modello CRISP-DM*



2.2.1 Business Understanding

In questa prima fase si pone l'attenzione all'identificazione delle principali aspettative di business del progetto ed è fondamentale comprendere gli obiettivi che si desiderano ottenere e verificare se il problema possa essere effettivamente affrontato con tecniche di Machine Learning per ottenere reali benefici.

2.2.2 Data Understanding

Vengono esaminati i dati messi a disposizione per le analisi, alla luce degli obiettivi di business che sono stati decisi durante la prima fase.

2.2.3 Data Preparation

Questa fase consiste nel ripulire, ordinare ed elaborare tali dati per renderli utilizzabili secondo gli scopi predefiniti. L'obiettivo è quello di preparare il miglior dataset possibile che prende il nome di *gold standard*. Quasi tutti gli algoritmi di Machine Learning si aspettano features o valori numerici in input, questo perché utilizzano tecniche di analisi matematica e differenziale che prevedono necessariamente di poter elaborare numeri. Per questo motivo durante la fase di *data preparation* ci vengono in aiuto alcune tecniche di *Encoding* che consentono di tradurre features non numeriche in un valore numerico o tecniche da utilizzare nel caso in cui nel nostro dataset siano presenti valori mancanti che potrebbero portare a *underfitting*. Questo processo prende il nome di *feature design* o *feature engineering* o più genericamente *dataset engineering*. Sostanzialmente consiste nel modellare i dati del dataset per far sì che non contengano informazioni superflue, ridondanti, ma funzionali e sufficienti, ricorrendo ad operazioni matematiche, filtri o semplicemente alla rimozione o aggiunta di features al dataset.

2.2.4 Modelling

Attraverso algoritmi di Machine Learning adeguati, si andrà a costruire e ad adattare il modello ottimale ai dati pre-elaborati (processo di *fitting*).

2.2.5 Evaluation

Tramite l'analisi dei risultati si valuta se sono stati raggiunti gli obiettivi prefissati e si ipotizza una futura applicazione del modello.

2.2.6 Deployment

Nella fase finale il modello sviluppato viene applicato con l'obiettivo di valutare se è in grado di generare un impatto positivo sull'attività. Inoltre occorre tener conto che i modelli vanno costantemente tenuti in allenamento sottoponendoli periodicamente a fasi di training periodico.

La fase di modelling implica una metodologia di apprendimento, la quale classifica tre macro-famiglie principali:

2.3 APPRENDIMENTO PER RINFORZO (Reinforcement Learning)

Nell'*Apprendimento per Rinforzo* gli algoritmi agiscono sulla base di ricompense, che vengono erogate in funzione degli obiettivi raggiunti. I modelli di *Reinforcement Learning* prevedono tipicamente l'utilizzo di tre tipologie distinte di logiche: la prima cerca di effettuare delle predizioni, la seconda si occupa di valutare questi output sulla base di alcuni obiettivi e fornire dei feedback, mentre la terza fase consiste nel riprogrammare il modello per migliorarlo sulla base dei feedback ricevuti [8].

2.4 APPRENDIMENTO NON SUPERVISIONATO (Unsupervised Learning)

La tecnica dell'*Apprendimento Non Supervisionato* si rivela utile nel caso in cui non si hanno a disposizione dati storici con risultati noti (*Labeled Data*). L'obiettivo dell'*Unsupervised Learning* è quello di analizzare i dati a disposizione per ricavare

informazioni utili a eseguire predizioni attraverso metodi induttivi. Questo avviene adottando principalmente una tecnica denominata *Clustering*: algoritmi di questo tipo vanno a elaborare i dati a disposizione per individuare delle logiche, correlazioni o pattern tra di loro, al fine così di suddividere i dati in differenti gruppi chiamati *Cluster*, utilizzando concetti di similitudine e correlazione tra gli elementi. Più nello specifico il *Clustering* si avvale di correlazioni individuate grazie a distanze calcolate tramite formule matematiche e ogni algoritmo utilizza la tipologia di distanza più appropriata alla risoluzione di un problema [8].

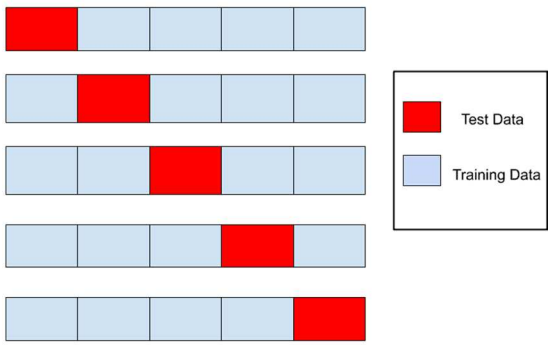
2.5 APPRENDIMENTO SUPERVISIONATO (Supervised Learning)

2.5.1 Training

Nel *Supervised Learning* durante la fase di apprendimento, denominata anche processo di training, si lavora su *Labeled Data*, i dati noti, per dimensionare i parametri e i coefficienti del modello di Machine Learning adottato. L'obiettivo è quello di ottenere la *learn function* ottimale e adattare al meglio il modello predittivo (*processo di fitting*). È buona pratica suddividere il dataset in *training* e *testing data*. Una parte dei dati di training saranno a loro volta dedicati alla validazione (*validation set*), cioè saranno utilizzati per reiterare molteplici volte il nostro algoritmo cambiando *hyperparameter* per minimizzare l'errore medio, tramite ad esempio la tecnica della *Cross Validation* e valutando le performance di molteplici modelli. La *Cross Validation* consiste nel suddividere il dataset in k parti e, a turno, si associa ciascuna di queste parti al *validation set* andando a fare poi le previsioni sui dati restanti. La *Cross Validation* è la soluzione ideale che sostituisce il campionamento asimmetrico il quale potrebbe generare un elevato

bias (scostamento medio tra valore predetto e valore reale), e inoltre riduce il rischio di *Overfitting* che potrebbe essere causato da numerosi *outliers*. In alternativa a questa tecnica, si ha la *Percentage Split* che si limita a dividere il dataset in due porzioni definite dall'operatore con una percentuale appunto. La decisione di utilizzare la tecnica della *Cross Validation* piuttosto che la *Percentage Split* è stata dettata dal fatto che quest'ultima necessita dataset di dimensioni maggiori per poter risultare efficace, perciò non è stata ritenuta adeguata nel caso in esame. La quota parte di *testing data* viene utilizzata durante una iterazione successiva, al fine di testare i risultati dell'algoritmo. Una volta terminate le fasi di *training* e *validation* sarà possibile utilizzare il modello al fine di predire risultati non noti a fronte di nuovi dati in input, ovvero si procede alla *fase di predizione o inferenza* [8].

Figura 2. *Cross-Validation*



2.5.2 Test e Accuratezza

Nella fase di *testing e accuratezza* viene utilizzata un'ultima parte del dataset per simulare il comportamento del modello come se gli output non fossero noti, quindi il dataset verrà fornito all'algoritmo ottimale senza i *Labeled Data*, rendendo quindi imparziali le predizioni dell'algoritmo. L'obiettivo è valutare l'accuratezza del

modello e per fare ciò il processo di *testing* prevede di far girare il modello una sola volta sui dati di test. È importante quindi distinguere la validazione, ovvero la fase in cui si cerca di ottimizzare le performance del modello, con la fase di *testing* dove invece si applica l'algoritmo e i parametri selezionati sui dati di set. Il *testing* utilizza l'algoritmo preselezionato per eseguire le predizioni, senza conoscere i dati di output: questo processo di predizione prende anche il nome di *inferenza*. Una volta ottenuti i risultati con l'operazione di *testing*, andremo a confrontarli con i dati noti per valutare l'accuratezza.

Può presentarsi il caso in cui nella fase di training si ottengono delle performance nettamente migliori rispetto alla fase di test e ciò potrebbe significare che il modello progettato lavori in *Overfitting*: nella fase di training l'algoritmo ha seguito fedelmente le osservazioni compresi i dati non rappresentativi, gli *outliers*, che distolgano l'attenzione da quelli effettivamente necessari. In antitesi all'*Overfitting* si ha *l'Underfitting*, condizione che si verifica quando il modello necessita di più informazioni e dati. In tal caso occorre ripetere l'intera procedura da capo e riformulare il modello ripartendo dalla fase di *training*, *validation* e quindi *test* finché i risultati non saranno soddisfacenti. Al contrario, se i risultati sono soddisfacenti e quindi il modello non è sbilanciato né in *Overfitting* né in *Underfitting*, allora esso è pronto per lavorare con osservazioni sul mondo reale.

2.6 REGRESSIONE E CLASSIFICAZIONE

Gli algoritmi di Machine Learning per l'addestramento supervisionato vanno a indirizzare normalmente due tipologie di problematiche: *Regressione* e *Classificazione*. Quando si tratta di predire valori numerici continui potenzialmente

infiniti si parla di *regression problem*, mentre se si tratta di predire l'appartenenza ad una classe o categoria si parla di *classification problem* [8,9].

2.6.1 Classificazione

La classificazione è forse la tecnica di Data Mining più comunemente applicata il cui obiettivo è quello di suddividere la problematica in classi per ridurre l'entropia finale, ovvero il livello di disordine del sistema. Viene utilizzata per analizzare grandi quantità di dati in modo automatico o semiautomatico ed estrarne conoscenza a livello di variabili categoriche o classi. Di solito, si ha a che fare con grandi quantità di dati (commerciali, finanziari, scientifici, clinici, ecc.) che possono anche essere:

• INCOMPLETI: mancano delle osservazioni;

• RUMOROSI: presentano dei valori anomali;

• INCONSISTENTI: esistono codici differenti per lo stesso item;

• RIDONDANTI: presenza della medesima informazione.

3 STRUMENTI

3.1 DESCRIZIONE DEL DATASET

Il lavoro è stato eseguito su un dataset dei reclami inizialmente contenente 69 colonne che riassumevano le informazioni riguardanti le caratteristiche fisiche del prodotto come ad esempio il colore, i materiali di cui è composto e il modello, le informazioni sul reclamo, come la data in cui è stato effettuato, e la descrizione del difetto stesso. I dati sono stati forniti in forma tabellare su un foglio di calcolo Excel

in formato CSV (*Comma Separated Values*), ma è stato necessario procedere ad una elaborata fase di preprocessing per ripulire il dataset da dati incompleti, rumorosi, inconsistenti e ridondanti.

Figura 3. *Alcune colonne del dataset a disposizione*

K	L	M	N	O	P	Q	R	S	T	U	V
Product Category	Product Line	Product Type	Style ID	Material ID	Color ID	Cites	MTO	Purchase Date	Lifespan	Launch Season in Retail	Probabilità
Mini Bags	Mon Trésor	Mini Bag	8BS010	A6V8	F0KUR		NO MTO	06/03/2019	Seasonal	S/S 2019	Main Body Other Materials Surface/Color Alteration
Bags	Shopper	Shopping	8BH348	A5K4	F15WY		NO MTO	31/03/2019	Seasonal	S/S 2019	Main Body Fabric Bubbling
SLG	F is Fendi	Long Wallet	8M0365	AAFM	F13VK		NO MTO	29/07/2020	Permanent	S/S 2019	Accessories Functional Accessories Broken
Mini Bags	Wallet On Chain F	Chain Pouch	8BS006	A6CA	F13WB		NO MTO	13/06/2019	Permanent	F/W 18-19	Main Body Leather Surface/Color Alteration
Mini Bags	Wallet On Chain F	Chain Pouch	8BS006	A5TY	F13WB		NO MTO	28/04/2019	Permanent	F/W 18-19	Main Body Leather Surface/Color Alteration
Bags	Peekaboo	Top Handle	8BN244	A85X	F184T	C	NO MTO		Seasonal	F/W 19-20	Main Body Leather Holes
Mini Bags	Wallet On Chain F	Chain Pouch	8BS006	A6CA	F13WB		NO MTO	13/11/2019	Permanent	F/W 18-19	Main Body Leather Surface/Color Alteration
Bags	Baguette	Flap Bag	8BR771	AAF2	F0MK5		NO MTO	16/10/2019	Seasonal	S/S 2020	Accessories Functional Accessories Improper Working
Bags	Peekaboo	Hobo	8BN304	A94O	F18MQ		NO MTO	26/11/2020	Seasonal	F/W 19-20	Handle Leather Marks/Scratches
Bags	Mon Trésor	Bucket	8BT298	A7QO	F17A4		NO MTO	14/04/2019	Seasonal	S/S 2019	Main Body Other Materials Surface/Color Alteration
SLG	F is Fendi	Long Wallet	8M0251	A6CB	F13VJ		NO MTO		Permanent	S/S 2019	Main Body Leather Surface/Color Alteration
SLG	FF	Phone Pouch	7AR675	A7TT	F14CG		NO MTO	27/12/2019	Cross-seasonal	S/S 2019	Accessories Functional Accessories Improper Working
Mini Bags	Wallet On Chain F	Belt Bag	8BM005	A6CB	F13VJ		NO MTO	01/12/2020	Cross-seasonal	S/S 2019	Accessories Leather Surface/Color Alteration
Mini Bags	Mon Trésor	Mini Bag	8BS010	A5PK	F0KUR		NO MTO	18/01/2019	Seasonal	S/S 2019	Accessories Ornamental Accessories Broken
Bags	Peekaboo	Top Handle	8BN310	AAFL	F19TX		NO MTO		Seasonal	S/S 2020	Accessories Functional Accessories Broken
Mini Bags	Mon Trésor	Mini Bag	8BS010	ABMR	F0A6E		NO MTO	01/09/2020	Seasonal	S/S 2020	Strap Fabric Loose thread
Bags	Baguette	Flap Bag	8BR600	A6V5	F17U4		NO MTO	14/05/2020	Permanent	F/W 19-20	Main Body Fabric Stitching
Bags	Mon Trésor	Bucket	8BT298	A7QO	F17A3		NO MTO		Seasonal	S/S 2019	Main Body Other Materials Surface/Color Alteration
Bags	Baguette	Flap Bag	8BR783	ACNZ	F1C0I		NO MTO		Cross-seasonal	F/W 20-21	Main Body Leather Marks/Scratches
Bags	FF	Messenger	7VA470	A80P	F0P0N		NO MTO	03/08/2019	Seasonal	F/W 19-20	Lining Leather Component not alligned
Bags	Boston	Boston	8BL137	NDU	F0JLX		NO MTO	25/12/2020	Permanent	S/S 2016	Accessories Functional Accessories Broken
Bags	Peekaboo	Top Handle	8BN290	Q0I	F0E66		NO MTO	21/09/2020	Permanent	F/W 15-16	Accessories Functional Accessories Broken
SLG	F is Fendi	Long Wallet	8M0365	AAII	F19DA		NO MTO		Permanent	S/S 2020	Lining Leather Stain
Bags	Shopper	Shopping	8BH360	A7S0	F0KUR		NO MTO	16/03/2019	Seasonal	S/S 2019	Main Body Other Materials Surface/Color Alteration
Bags	Baguette	Flap Bag	8BR600	A6V5	F17U4		NO MTO	13/06/2020	Permanent	F/W 19-20	Strap Other Materials Loose thread
SLG	Signature	Continental Wallet	7M0264	AFCL	F0GXN		NO MTO		Cross-seasonal	S/S 2021	Main Body Leather Marks/Scratches
Bags	Shopper	Shopping	8BH348	A5K4	F15WY		NO MTO	21/06/2019	Seasonal	S/S 2019	Handle Leather Bubbling
SLG	F is Fendi	Small Wallet	8M0395	A18B	F0KUR		NO MTO		Permanent	S/S 2018	Accessories Functional Accessories Broken
SLG	F is Fendi	Long Wallet	8M0365	ACOJ	F15KR		NO MTO	22/12/2020	Seasonal	F/W 20-21	Accessories Functional Accessories Broken
SLG	Baguette	Long Wallet	8M0365	AAJD	F19T7		NO MTO		Cross-seasonal	S/S 2020	Accessories Functional Accessories Improper Working
Bags	Peekaboo	Pochette	8BP118	AB5Y	F082F		NO MTO	14/08/2020	Seasonal	S/S 2020	Accessories Functional Accessories Improper Working

Essendo di fronte ad un problema di Data Science, è stato affrontato come tale, suddividendolo nelle fasi del CRISP-DM model esplicate precedentemente.

3.2 PARAMETRI DEL CRISP

- *Business Understanding*: un produttore necessita di un modello che sia in grado di classificare l'entità del difetto dei suoi articoli sulla base di informazioni relative alla provenienza dell'articolo, al venditore e alla descrizione del difetto, fornendo in input una base di dati su cui effettuare le analisi;
- *Data Understanding*: si hanno a disposizione due fogli di calcolo in cui il primo riassume le informazioni relative ad ogni articolo difettoso, mentre il

secondo contiene delle descrizioni più dettagliate delle colonne del primo foglio di calcolo;

- *Data Preparation*: questa è la fase che richiede il maggior dispendio di lavoro e tempo. Analizzando il database è evidente che sono presenti dati che possono costituire un problema durante l'applicazione degli algoritmi di Machine Learning. Per evitare che ciò accada bisogna gestire tutti gli *outliers*, ovvero quei dati che non servono alla risoluzione del problema. Seppur nel software utilizzato per l'analisi del dataset siano disponibili numerosi filtri per il *preprocessing* che consentono di operare sui record del dataset e sulle colonne, inizialmente è stato più agevole lavorare direttamente su Excel. Tenendo in considerazione il significato dei vari campi del primo foglio di calcolo, è stato snellito il database riducendo il numero delle colonne a 25, eliminando tutte quelle ritenute superflue e ininfluenti alla problematica. In secondo luogo sono stati eliminati alcuni record che contenevano dati mancanti, eccetto nelle colonne in cui erano in misura maggiore rispetto ai campi compilati. La decisione di eliminare un dato piuttosto che adottare delle tecniche di manipolazione per poterlo reintegrare correttamente nel database, è stata dettata dal fatto che in rapporto alla mole di informazione presente nella base di dati non avrebbe alterato eccessivamente la struttura del dataset. Durante la fase di *preprocessing* dei dati bisogna inoltre verificare che le colonne non contengano dati ridondanti o erroneamente registrati. Nel caso in cui l'errore sia circoscritto in poche celle è possibile intervenire manualmente, altrimenti se il numero delle celle dovesse essere cospicuo l'operazione di aggiustamento verrebbe agevolata dall'uso di filtri. Nel caso specifico

vengono riportate alcune delle operazioni di eliminazione delle features dalla base di dati:

- La colonna *ID* è un identificativo che non incide sulle caratteristiche dell'articolo in questione perciò non rappresenta un'informazione utile per predire o classificare eventualmente un articolo difettoso;
- Le colonne *SKU2* e *StyleMatID* contengono un accorpamento di n codici che sono già registrati in altre n colonne esclusivamente dedicate, perciò si tratta di dati ridondanti.

La fase di *preprocessing* implica, se necessario, anche l'aggiunta di features ritenute utili: è il caso di *Periodo Trascorso* che calcola, grazie a una formula elaborata in Excel, il numero di giorni intercorsi tra la data di acquisto del prodotto e quella di presentazione del reclamo.

Viene riportata qui di seguito una descrizione dettagliata delle colonne che compongono il dataset ultimato e pronto alla fase di *Modelling*:

•*New Zone*: indica la zona geografica di apertura del reclamo;

•*Stock Origin*: identifica chi ha aperto il reclamo, Stock (negozio) oppure Client (cliente);

•*Classification defect main location*: livello 1 di classificazione del reclamo. È una colonna con 5 attributi che identificano la componente della borsa che risulta essere difettosa (maniglia, cinghia o main body);

•*Classification defect detailed location*: indica cosa risulta essere difettoso del componente descritto dal primo livello di classificazione, se si tratta quindi del pellame o di un difetto di produzione o altri accessori ornamentali;

•*Classification defect type*: contiene una descrizione più accurata del difetto;

•*Product Category*: Descrive la categoria di appartenenza del prodotto difettoso (Mini Bags, SLG, Bags);

•*Product Line*: descrive la linea di produzione dell'articolo;

•*Product Type*: descrive in modo dettagliato la tipologia del prodotto. Nel dataset troviamo 29 diversi tipi di prodotti;

•*Style Id*: contiene un id identificativo dello stile del prodotto;

•*Material Id*: è l'identificativo del materiale con cui è stato prodotto l'articolo;

•*Color Id*: è l'identificativo del colore del prodotto;

•*Cites*: indica se è presente o meno una documentazione relativa all'utilizzo di pellame pregiato: *c* se è *coccodrillo* e *d* se è *declaration*;

•*MTO*: *Make to order* contiene due campi che sono *NO MTO*, *MTO*. Sono solo 11 i record in cui compare *MTO*: in effetti è inusuale che un prodotto fatto su misura esca dalla catena di produzione difettoso;

•*Purchase Date*: indica la data di acquisto del prodotto;

•*Created Date*: indica la data in cui è stato presentato il reclamo relativo all'articolo acquistato in *Purchase Date*;

•*Periodo trascorso*: contiene il numero di giorni trascorsi tra la data di acquisto dell'articolo e la data in cui è stato presentato il reclamo;

•*Lifespan*: periodo di disponibilità (permanent/seasonal/cross-seasonal)

•*Probabilità*: descrizione dettagliata del difetto;

•*Descrizione parte*: descrizione della parte difettosa;

•*Material Type*: descrizione nominale del materiale che compone l'articolo. Si hanno 27 tipologie diverse di materiali;

•*Tipo Materiale*: a differenza di *Material Type* descrive solo il materiale che compone per la maggiore l'articolo mentre *Material Type* tiene conto anche di composizioni miste di materiali;

•*Probabilità Rischio per Modello Parte*: contiene valori numerici da 1 a 3 per descrivere la probabilità che la parte possa essere la causa del difetto;

•*Matching*: contiene valori numerici da 1 a 6;

•*Matching Probabilità Modello Parte+ Gravità Difetto*: contiene i valori di *ALTO*, *MEDIO*, *BASSO*, valutati facendo un matching tra la *Probabilità Rischio per Modello Parte* e la gravità del difetto. È la *Target Label*, la feature che deve essere classificata.

A questo punto il dataset è ancora soggetto alla fase di *preprocessing*, in cui verranno applicati ulteriori filtri attraverso il software in cui è stato importato.

È opportuno convertire quanti più attributi nominali in numerici poiché semplifica le operazioni di calcolo agli algoritmi di Machine Learning. Partendo dalla prima, andremo ad analizzare una ad una le features e le labels che esse contengono, valutando se è possibile convertire i valori della feature da nominali a numerici e se è necessario applicare ulteriori filtri. Il filtro di cui ci serviremo è un *NominalToBinary* che converte gli attributi nominali in binari. Applicare un filtro *NominalToBinary* ad attributi che contengono n labels significa che la colonna viene esplosa in n colonne, dove ciascuna colonna contiene il valore *0* se il record non è associato a quella label, il valore *1* se appartiene. È sconsigliato utilizzare questo filtro dove le labels superano un certo valore di n perché il dataset si popolerà di eccessive colonne e potrebbe andare in *Overfitting*. In questo caso ho ritenuto opportuno applicare il *NominalToBinary* fino ad un massimo di 5 labels:

Stock Origin (Nominal), *Classification defect main location*, C*lassification defect detailed location*, *Product Category, Cities, MTO* e *Lifespan* sono le features nominali a cui è stato applicato il filtro per consentire al modello di interagire in modo più efficiente con il dataset. Una volta completata la fase di *preprocessing* dei dati, è consentito procedere alla fase successiva, quella di *modelling*. In questa fase, prima di applicare ai dati degli algoritmi di Machine Learning, occorre suddividere il dataset in due parti: una dedicata al *training* e *validation* mentre l'altra alla fase di *test*. Il software utilizzato nel progetto consente di utilizzare delle Test Option che agevolano la ripartizione del dataset con lo scopo di evitare di andare in *Overfitting* durante l'elaborazione dei dati.

3.3 WEKA

Weka (Waikato Environment for Knowledge Analysis) è un software sviluppato dall'università di Waikato in Nuova Zelanda, fondato da Ian Witten nel 1992 la cui interfaccia grafica presenta [5,16]:

Figura 4. *Ambiente di lavoro Weka*

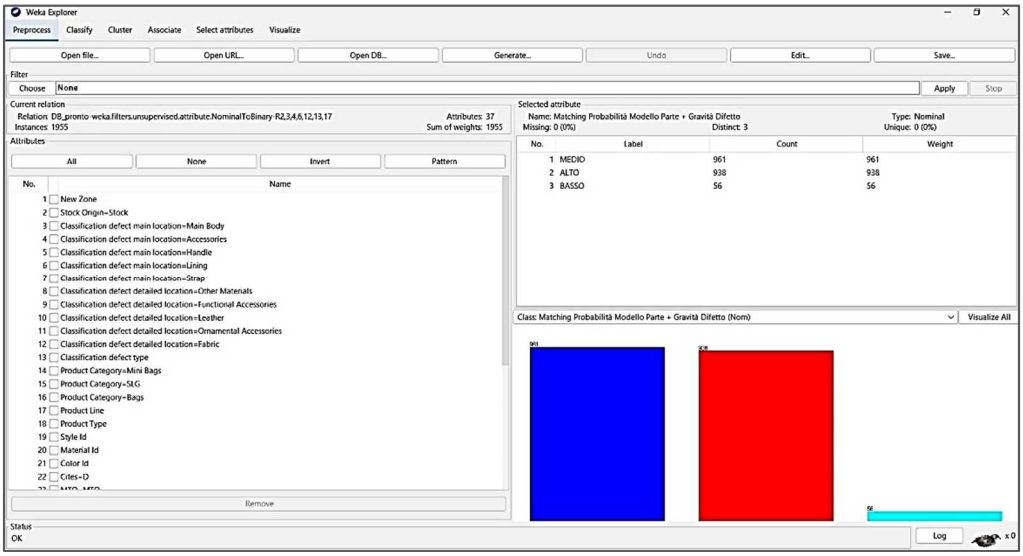
•*Simple CLI*: permette l'esecuzione di programmi in linea di comando, da cui appunto CLI (*Command Line Interface*). Tale modalità simula l'esecuzione dell'algoritmo prescelto come fosse un programma a parte;

•*Explorer*: rappresenta l'interfaccia grafica di riferimento per la fase di analisi. Si presenta all'utente come un insieme di pagine sovrapposte, ciascuna delle quali offre le varie fasi di analisi: la preparazione dei dati, la classificazione, il clustering, l'associazione, la selezione automatica degli attributi e la visualizzazione grafica dei dati;

O Nella prima pagina *Preprocess*, l'utente è guidato nella selezione dei dati da utilizzare per l'analisi: le fonti supportate sono file locali e remoti (nei formati ARFF, CSV) oppure tramite l'esecuzione di una query di selezione su un

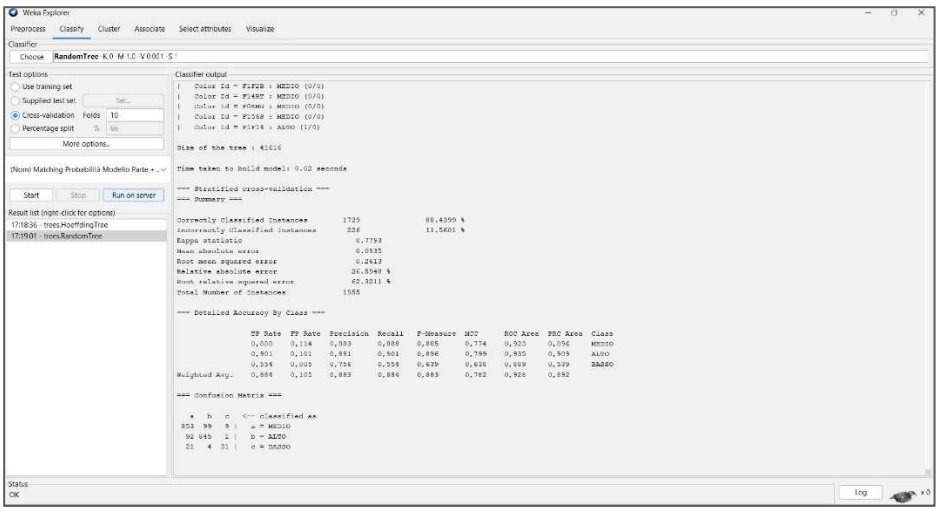
datatabase. Una volta importati i dati, il dataset è pronto per essere eventualmente modificato in modo manuale o automatico mediante l'applicazione di filtri. La pagina è composta da tanti pannelli con le informazioni più rilevanti del dataset di analisi: la numerosità, il numero di frequenze e la distribuzione di ciascun attributo;

Figura 5. *Pannello Preprocess*



O *Classify*: serve per applicare ai dati gli algoritmi di classificazione e i modelli per la regressione (entrambi sono chiamati *classifiers* in Weka);

Figura 6*. Pannello Classify*



O *Cluster*: permette di utilizzare tecniche di *cluster analysis*;

O *Associate*: serve per applicare algoritmi di apprendimento delle regole di associazione, ovvero relazioni nascoste tra i dati;

o *Select Attributes*: esegue degli algoritmi che permettono di valutare gli attributi in base alla loro utilità per la classificazione;

o *Visualize*: si propone come strumento per l'analisi grafica: grafici bidimensionali, detti anche *scatter plots* o *scatter graphs*, cioè un tipo di grafico in cui due variabili di un set di dati sono riportate su uno spazio cartesiano, ottenuti mediante tutte le possibili combinazioni tra le variabili con la possibilità di effettuare zoom e selezionare singole unità presenti in ciascun grafico;

•*Experiment*: questa interfaccia grafica si propone come soluzione di rapida configurazione al problema dell'esecuzione sincrona di più esperimenti: l'obiettivo è quello di applicare contemporaneamente diversi algoritmi di data mining su uno stesso dataset in ingresso e di poter analizzare successivamente i risultati ottenuti;

•*KnowledgeFlow*: è uno strumento grafico per la modellazione del cosiddetto data-flow, ossia la sequenza di operazioni successive di cui si compone l'analisi, il cui obiettivo consiste nel favorire un approccio sistematico al problema;

•*ArffViewer*: è uno strumento per la visualizzazione e la modifica dei dati provenienti da un file di tipo ARFF, il formato nativo di Weka.

L'Attribute Relation File Format è un file CSV preceduto da un'intestazione contenente delle informazioni sui dati. Il file inizia con una riga contenente il tag @relation, che indica il nome o la descrizione del dataset. Seguono tante righe precedute dal tag @attribute quanti sono gli attributi di ciascuna osservazione: per ogni attributo è specificato il nome e il tipo. La sezione dedicata alle osservazioni è segnalata dalla riga con il tag @data.

Figura 7. *Esempio di file strutturato in ARFF*

```
@relation airline_passengers
@attribute passenger_numbers numeric
@attribute Date date 'yyyy-MM-dd'

@data
112,1949-01-01
118,1949-02-01
132,1949-03-01
129,1949-04-01
121,1949-05-01
135,1949-06-01
148,1949-07-01
148,1949-08-01
136,1949-09-01
```

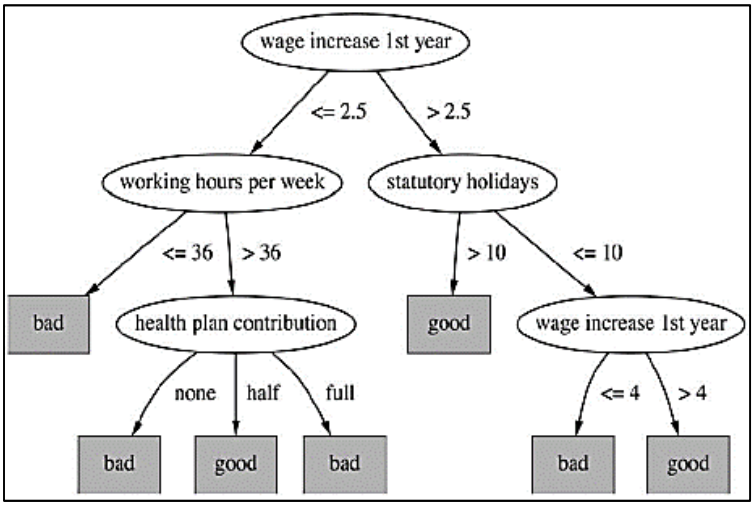
3.4 ALGORITMI

3.4.1 Classification tree

I *Classification Tree* sono classificatori con apprendimento supervisionato. In questo caso le foglie del nostro albero di decisione ci presenteranno delle predizioni relative a classi o categorie. L'obiettivo principale del classificatore è quello di suddividere le osservazioni in maniera da ridurre l'entropia o livello di disordine man mano che si va avanti, quindi man mano che il *Decision Tree* passa dal livello superiore al livello inferiore. Perciò ad ogni livello si sceglie la feature che massimizza il cosiddetto *information gain*, che consente di confrontare quanta

entropia viene sottratta tra il livello padre e quello figlio, ovvero la capacità che ogni feature ha di separare un certo numero di classi. Nel *Classification Tree* le features con la maggiore quantità di informazione devono essere controllate dai nodi più vicini alle radici e via via i nodi più lontani lavoreranno su features che qualificano meno il campione [8].

Figura 8. *Esempio di Classification Tree*



3.4.2 ZeroR

Il Classificatore *ZeroR* assegna tutte le istanze alla classe di maggiore dimensione presente nel training-set senza mai considerare gli attributi di ciascuna istanza. Non ha molto senso usare questo schema per la classificazione, per cui gli esperimenti eseguiti con *ZeroR* servono solo come parametro di riferimento per la valutazione della performance degli altri classificatori, poiché dovrebbe rappresentare il peggior risultato possibile. Predice la media (per una classe numerica) o la moda (per una classe nominale) [17].

3.4.3 Decision Stump

E' un modello che partendo dalla *root* si dirama immediatamente nelle *leaf* (foglie) basandosi sul valore di una sola feature in input, ritenuta la più rilevante [4].

3.4.4 Hoeffding Tree

L'*Hoeffding Tree* è un albero decisionale che sfrutta il *limite di Hoeffding* per dimostrare che un piccolo campione può essere sufficiente a scegliere un attributo di suddivisione ottimale delle varie features. Alle origini l'*Albero di Hoeffding* veniva utilizzato per tenere traccia dei flussi di clic nel web e costruire modelli che potessero prevedere la probabilità di accesso di un utente ad un certo host e sito web. L'algoritmo utilizza il limite (di Hoeffding) per determinare, al momento della selezione di un attributo in un nodo, il numero più piccolo di esempi necessari per suddividere il nodo stesso [7].

3.4.5 Random Tree

Questo albero decisionale introduce una casualità: piuttosto che ricercare la migliore feature come condizione per suddividere un nodo, trova la migliore feature in un sottoinsieme casuale dell'insieme globale delle feature stesse [9].

3.4.6 Random Forest

L'algoritmo *Random Forest* è un insieme di *Decision Trees* poiché rientra nell'*ensemble learning*, la categoria di algoritmi che usano più algoritmi di Machine Learning per ottenere predizioni più accurate. Per questo motivo minimizza *l'Overfitting*, ovvero il rischio di adattarsi troppo fedelmente ai dati durante la fase

di Training. Il numero di alberi dipende dalla natura del set di training e da altri parametri come il numero di classi e la *max_depth* e può essere trovato eseguendo l'ottimizzazione degli *hyperparameter*. La struttura del *Random Forest* implica una maggior diversità dei singoli alberi decisionali che lo compongono [1].

3.5 INDICI DI VALUTAZIONE

A seguito dell'applicazione degli algoritmi occorre effettuare un'analisi dei risultati per capire quale classificatore meglio rispecchia le esigenze e le specifiche del problema. Si andranno ad analizzare i seguenti indici con particolare attenzione:

3.5.1 K-Statistic

Il *K-Statistic* è un indice di accuratezza che confronta una *Observed Accuracy*, ovvero il numero di istanze classificate correttamente, con una *Expected Accuracy*, che indica di quanto il numero di istanze classificate si avvicina al valore reale. Viene utilizzato non solo per valutare un singolo classificatore, ma anche per valutare i classificatori tra di loro[10]. Basandosi sulla *Confusion Matrix*, la quale sulle righe presenta i valori reali e corretti mentre sulle colonne i valori predetti dall'algoritmo (o viceversa), definiamo quindi:

- *Observed Accuracy* (PA) è il numero di istanze classificate correttamente nell'intera matrice di confusione;

- *Expected Accuracy* (PE) indica quanto il numero di istanze classificate si avvicina al valore reale di quella quantità;

Equazione 1. *Formula Indice K*

$$K = \frac{P(A) - P(E)}{1 - P(E)}$$

Figura 9*. Matrice di Confusione*

	Cats	Dogs
Cats	22	9
Dogs	7	13

- **Ground truth**: Cats (29), Dogs (22)
- **Machine Learning Classifier**: Cats (31), Dogs (20)
- **Total**: 51
- **Observed Accuracy**: ((22 + 13) / 51) = 0.69
- **Expected Accuracy**: ((29 * 31 / 51) + (22 * 20 / 51)) / 51 = 0.51
- **Kappa**: (0.69 – 0.51) / (1- 0.51) = 0.37

L'interpretazione del *K-Statistic* dipende dal contesto in cui ci si trova ma in comune accordo si ha a disposizione una scala che riassume i range di accettabilità dell'indice K. In modo approssimativo possiamo ritenere la concordanza (k) [15]:

- K <= 0 scarsissima
- K = 0.01 – 0.20 scarsa
- K = 0.21 – 0.40 discreta
- K = 0.41 – 0.60 moderata
- K = 0.61 – 0.80 buona
- K = 0.81 – 1.00 ottima

Più il grado di accordo è maggiore, maggiore sarà il valore dell'indice e quindi l'affidabilità fornita dalla classificazione.

3.5.2 Mean Absolute Error (MAE)

Il rapporto tra la somma degli errori calcolata in valore assoluto e il numero di previsioni effettuate, cioè la media. Perciò il modello in esame è migliore tanto più il valore dell'indice è prossimo allo zero [14].

3.5.3 Root Mean Squared Error (RMSE)

La cosiddetta deviazione standard, cioè la radice della media dei quadrati degli scostamenti tra il valore vero e il valore predetto. Questo parametro fornisce un'idea di quanto il sistema, con la sua predizione, si allontana dalla realtà dei dati. È un indice di accuratezza, la quale indica quanto una misura è vicina al valore accettato, perciò descrive una proprietà del risultato [14].

3.5.4 Relative Absolute Error (RAE)

Confronta l'errore di previsione effettivo con l'errore di previsione di un modello semplicistico (naive). Un buon modello di previsione produrrà un rapporto prossimo allo zero, mentre un modello scadente (uno peggiore del modello ingenuo) produrrà un rapporto maggiore di uno. È un indice di precisione, il quale quantifica il grado di efficacia con cui sono state effettuate le misure, o l'adeguatezza dei calcoli effettuati. La precisione quindi fornisce un'informazione riguardante il processo di misurazione e non sul risultato stesso [13].

4 METODOLOGIA E RISULTATI

4.1 TABELLA DEI RISULTATI

Tabella I. Tabella dei Risultati

	Zero R	Decision Stump	Hoeffding Tree	Random Tree	Random Forest
Kappa Statistic	0	0,9443	0,8863	0,7793	0,9511
MAE	0,3517	0,0361	0,0404	0,0935	0,0917
RMSE	0,4193	0,1343	0,1845	0,2613	0,1583
RAE (%)	100	10,2611	11,4899	26,5948	26,0741
Correctly Classified Instances (%)	49,156	97,1355	93,9642	88,4399	97,4425
Incorrectly Classified Instances (%)	50,844	2,8645	6,0358	11,5601	2,5575

4.2 INTERPRETAZIONE DEGLI INDICI

Dalla tabella mostrata si può notare come il valore più soddisfacente di *K-Statistic* si ottenga in corrispondenza del *Random Forest*, il quale, rientrando nella categoria degli algoritmi di *ensamble learning*, minimizza il rischio di incorrere in *Overfitting*, garantendo perciò una maggiore affidabilità e accuratezza dei risultati.

Proseguendo la valutazione dei risultati, è evidente che il *Decision Stump* offre dei valori competitivi sia per quanto riguarda il *Mean Absolute Error* (MAE), che il *Root Mean Squared Error* (RMSE) che il *Relative Absolute Error* (RAE).

Per poter commentare correttamente l'esito delle varie applicazioni degli algoritmi è necessario tenere in considerazione sia la struttura del modello applicato che la dimensione del dataset che si ha a disposizione. La base di dati che riassume le informazioni dei reclami, in seguito alla fase di *pre-processing*, può essere ritenuta esigua rispetto alle consuete moli di dati che vengono solitamente elaborate dagli algoritmi in esame.

Proseguendo infatti con l'analisi dei risultati, la correlazione che persiste tra la dimensione del dataset e la complessità dell'algoritmo, trova corrispondenza nel valore del RMSE, meglio noto come deviazione standard. Ricordando che è un indice di precisione che indica di quanto il sistema si allontana dalla realtà e di quanto le misurazioni effettuate ricadano in un intervallo ridotto, è il *Decision Stump* a presentare il valore più basso e di conseguenza più vantaggioso. Questo modello infatti adotta un solo criterio di decisione per selezionare e classificare le varie osservazioni ed essendo quindi influenzate da un solo parametro sono soggette a meno varianza e ricadono in un intervallo ridotto.

Il *Relative Absolute Error* fornisce un'idea di quanto il classificatore sia affine al dataset a cui viene applicato. Per affinità si intende la propensione del modello a sfruttare le proprie potenzialità e caratteristiche in relazione alla base di dati con cui interagisce. A conferma di ciò si può notare come il *Decision Stump* presenti ancora il valore più soddisfacente tra quelli ottenuti dai classificatori in esame, mentre il *Random Forest*, che è un modello molto più elaborato e complesso, è caratterizzato da un RAE di gran lunga più elevato rispetto agli altri classificatori, proprio perché la complessità del modello è sproporzionata a quella del dataset in esame.

In sintesi algoritmi di una esigua complessità strutturale si adattano meglio a dataset di dimensioni ridotte mentre i modelli più complessi lavorano meglio su dataset estesi. È probabile che queste considerazioni siano in controtendenza rispetto alle aspettative che sarebbero potute essere riposte nel modello del *Random Forest* per via delle sue potenzialità strutturali: come si può notare infatti, nonostante il *Random Forest* abbia di gran lunga una struttura assai più articolata rispetto agli altri classificatori elencati, i risultati ottenuti non differiscono di molto da quelli degli altri classificatori. È opportuno ribadire che ciò è dovuto alle dimensioni ridotte del dataset a disposizione che consente anche a modelli semplicistici come il *Decision Stump* di offrire prestazioni vantaggiose.

5 CONCLUSIONI E SVILUPPI FUTURI

In conclusione, seppur il *Decision Stump* si sia dimostrato un classificatore più che opportuno ed efficiente, ritengo doveroso considerare più affidabile l'utilizzo del *Random Forest* per via della sua duttilità. Per poter ottenere dei risultati ancora più attendibili da poter essere utilizzati in delle decisioni aziendali di un certo spessore, è necessario ampliare la base di dati che riassume le informazioni relative ai reclami. Non solo aggiungendo delle osservazioni, ma studiando accuratamente quali potrebbero essere delle ulteriori informazioni utili all'identificazione e alla classificazione dei difetti delle borse. Auspicando perciò ad un ampliamento del dataset, il *Random Forest* garantirà in futuro delle prestazioni attendibili fronteggiando anche una complessità del dataset maggiore.

Suggerisco di gestire in modo differente le varie features che contengono valori nominali. È preferibile utilizzare un valore alfanumerico piuttosto che nominale per molteplici motivi: in primo luogo è già stato riscontrato che gli algoritmi matematici di cui sono provvisti i modelli sono più efficienti se lavorano con valori numerici e inoltre agevola l'inserimento da parte dell'operatore che compila manualmente i campi del dataset, riducendo il rischio di incorrere in dati sporchi dovuti a una incorretta trascrizione. Infine il codice alfanumerico presenta un dataset più leggibile e ne facilita la consultazione. In conclusione il tutto si traduce in un focus maggiore alla fase di *data understanding* e ad un corposo processo di reperimento dei dati. Avere a disposizione un dataset di dimensioni maggiori e accuratamente definito aumenta sicuramente la probabilità di ottenere risultati ancora più precisi e affidabili, che in un business context come quello preso in esame, può sfruttare e mettere in risalto le potenzialità dell'Artificial Intelligence e del Machine Learning.

6 RIFERIMENTI

1. Arena, M. (s.d.). *Alberi Decisionali e Random Forest come funzionano*. Tratto da Quora: https://it.quora.com/Alberi-decisionali-e-random-forest-come-funzionano

2. Audrey Aurelia Chang, C. D. (2021). Fashion Trend Forecasting Using Machine Learning Techniques: A Review. *Data Science and Intelligence System*, 11.

3. Chakraborty, S. (2020). A comprehensive review on image based style prediction and online fashion recommendation. *Jomard Publishing*, 22.

4. Cividini, S. (s.d.). *TopoMirtillo.* Tratto da diligender.libero.it: https://digilander.libero.it/TopoMirtillo/DataMining.pdf

5. Fabio Bertozzi, G. C. (s.d.). Weka Data Mining System. *SIstemi informativi a supporto delle decisioni*. Bologna, Italia.

6. Gianpaolo Pigliasco, G. Z. (2009, Gennaio). *Wekametro*. Tratto da mokabyte: http://www.mokabyte.it/2009/01/wekameteo-1/

7. Ginni. (2021, Novembre 25). *What is Hoeffding Tree Algorithm*. Tratto da Tutorials Point: https://www.tutorialspoint.com/what-is-hoeffding-tree-algorithm

8. Gosmar, D. (2022). *Machine Learning: il sesto chakra dell'intelligenza artificiale.* Amazon.

9. Ibm. (2021, Aprile 09). *Random Trees node*. Tratto da Ibm.com: https://www.ibm.com/docs/en/cloud-paks/cp-data/3.5.0?topic=modeling-random-trees-node

10. Pozzolo, P. (2021, Aprile 10). *kappa cohen*. Tratto da paolapozzolo.it: https://paolapozzolo.it/kappa-cohen/

11. Radhika Pillarisetty, P. M. (2022). A Review of AI (Artificial Intelligence) Tools and Customer Experience in Online Fashion Retail. *International Journal of E-Business Research*, 12.

12. Roggero, C. (2022, Febbraio 27). *Danno di Immagine Azinedale*. Tratto da dandi.media: https://www.dandi.media/danno-di-immagine-aziendale/

13. Stephanie. (2019, Aprile 17). *relative absolute error*. Tratto da Statistichowto: https://www.statisticshowto.com/relative-absolute-error/#:~:text=Relative%20Absolute%20Error%20(RAE)%20is,clocks%2C%20rulers%2C%20or%20scales.

14. Tedesco, D. (2021, Maggio 26). *Dummies*. Tratto da Mediamenteconsulting.it: https://blog.mediamenteconsulting.it/2021/05/26/mm041/

15. Twain, J. (2017, Marzo 20). *Cohen's Kappa in plain English*. Tratto da StackExchange: https://stats.stackexchange.com/q/82162

16. Zamperin. (2007). Strumenti Open Source per Data Mining.

17. *Zeror the Simplest Possible Classifier*. (2020, Aprile 28). Tratto da R-Bloggers: https://www.r-bloggers.com/2020/04/zeror-the-simplest-possible-classifier-or-why-high-accuracy-can-be-misleading/

7 RINGRAZIAMENTI

Al termine dell'elaborato ci tengo a dedicare alcune righe ai ringraziamenti, per dimostrare la mia riconoscenza nei confronti delle persone che mi hanno accompagnato durante questo percorso di studi. Ringrazio:

Il mio relatore il Professor Zingaretti Primo, e i miei tutor Paolanti Marina e Pietrini Rocco per la professionalità e la disponibilità, oltre agli innumerevoli consigli utili alla realizzazione di questo progetto.

I miei amici e i miei colleghi, la cui complicità è sempre stata una risorsa importante.

La mia famiglia, i miei genitori Luca e Livia e i miei tre fratelli, Agnese, Francesco e Margherita che nell'arco di questi tre anni di studi mi hanno costantemente supportato e sopportato, come sempre del resto. La loro fiducia è per me indispensabile.

I miei nonni Ada e Germano, i cui sorrisi valgono più di mille parole.

La mia fidanzata Lucrezia, che giorno dopo giorno mi incoraggia a migliorare e a concentrarmi sui miei obiettivi.

Dedicata a mio nonno Germano.

Il tuo "Ingegnere"…