ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

DIPARTIMENTO DI INFORMATICA – SCIENZA E INGEGNERIA
Corso di Laurea in Ingegneria e Scienze Informatiche

APPLICAZIONE DI TECNICHE AI PER LA PROGETTAZIONE DI UN SISTEMA A SUPPORTO DEL PAZIENTE IPERTESO

Elaborato in
SISTEMI EMBEDDED ED INTERNET-OF-THINGS

Relatore
Prof. ALESSANDRO RICCI

Presentata da
MATTEO CASTELLUCCI

Corelatori
Prof.ssa SARA MONTAGNA
Dott. MARTINO PENGO

Anno Accademico 2019 – 2020

A Matteo, grazie per tutto quello che fai

Indice

Introduzione

L'Italia è un paese che invecchia, come molti di quelli che fanno parte del cosiddetto "occidente". Per questo motivo, le malattie che colpiscono gli anziani hanno un ruolo sempre più preponderante in medicina, sia per quanto riguarda la loro cura che in generale il loro studio. Una tra le più importanti, proprio perché tra le più diffuse, è l'ipertensione. Compito degli specialisti medici è quindi quello di elaborare protocolli sempre più efficaci per quanto riguarda la sua prevenzione e la sua cura. Queste procedure però cambiano nel tempo. Non tanto perché cambino i sintomi che contraddistinguono la malattia o gli effetti che ha nella vita dei malati, ma perché cambia la conoscenza che si ha di essa. Vengono scoperte cure o metodologie di diagnosi più efficaci, si definiscono meglio i parametri che permettono di dire se una persona ne soffre oppure no e così via. Quindi, periodicamente, vengono rivisti i metodi con i quali si individuano e si curano i malati di ipertensione. Si presenta perciò la necessità di individuare con quanta più precisione possibile le persone che soffrono di questa malattia senza saperlo, tra le altre cose, in modo da offrire loro una cura tempestiva che non peggiori il loro tenore di vita.

Un altro problema che l'ipertensione porta con sè è che non sempre le cure che vengono prescritte sono efficaci. Talvolta questo è dovuto al medico che non viene avvisato o non si accorge tempestivamente dei cambiamenti del decorso della malattia nel paziente. Altre volte invece è dovuto ad una certa tendenza alla negligenza da parte dei pazienti, che si sentono in buona salute, perché magari non hanno dei sintomi particolarmente significativi, e tendono a sottovalutare la loro malattia. Si rende perciò necessario fornire ai pazienti degli strumenti che li aiutino direttamente nella cura della loro malattia in maniera tale che sia per loro più facile curarsi. Questi ultimi però devono avere anche lo scopo di aiutare il medico nel suo lavoro di monitoraggio periodico delle condizioni di salute del paziente, perché possa avere realmente il polso della situazione.

Questo è quello che è emerso da alcuni incontri fatti con il dottor Martino Pengo dello "Istituto Auxologico Italiano" di Milano. Egli è parte di una commissione di medici dediti alla stesura dell'aggiornamento delle linee guida riguardanti l'ipertensione e lavora a stretto contatto con i pazienti che soffro-

no di questa malattia. Per questi motivi, a buon diritto, ci ha sottoposto i problemi precedentemente elencati.

Le soluzioni sono state ottenute grazie a quei campi di studio avanzati come l'intelligenza artificiale e l'*Internet of Things*, che proprio per le loro capacità stanno creando tecnologie sempre più pervasive. Per quanto riguarda l'AI, e in particolar modo il *Machine Learning*, stanno diventando sempre di più gli strumenti che mette a disposizione capaci di effettuare predizioni incredibilmente precise di fenomeni di qualsiasi natura. Mentre invece per quello che riguarda l'*Internet of Things*, esso sta rendendo un numero sempre maggiore di "Things", ma anche di compiti e processi, *smart*, quindi capaci di nuove e migliori funzionalità. Benché i contributi dati da ciascuno di questi due campi al mondo dell'*healthcare* sono significativi di per sè, essi hanno molti punti di contatto. Uno tra questi sono gli assistenti virtuali, altresì detti "chatbot". Essi sono dei prodotti dell'*Artificial Intelligence*, dato che trovano la loro *raison d'être* nel "Natural Language Understanding", ovverosia nell'estrazione di dati e conseguente comprensione di informazioni a partire da frasi complesse espresse in linguaggio naturale. Essi sono anche detti "agenti conversazionali" dato che il loro compito è essenzialmente quello di instaurare una conversazione con l'utente, che può fare loro richieste. Una naturale estensione di questi sistemi è quella di fare da controllori alle *Things* a cui possono essere connessi. Essi si pongono infatti come interfacce alternative agli ormai noti pulsanti e levette che fanno da pannello di controllo per i vari oggetti *smart*.

Gli argomenti appena citati sono stati quindi il *fil rouge* di questa tesi. In particolare, il primo progetto è consistito in un'analisi dei dati sul *dataset* ottenuto a partire dall'elaborazione delle risposte date ai questionari che sono stati distribuiti durante la "Giornata Mondiale dell'Ipertensione". Questi questionari contenevano domande sulla salute del paziente, assieme a domande di cultura generale, ed erano accompagnati da tre misurazioni di pressione arteriosa e frequenza cardiaca. Questo ha permesso di individuare il vero gruppo di interesse dell'analisi, che erano i non ipertesi noti, ovvero quelle persone che non soffrono o non sanno di soffrire di ipertensione e sono state categorizzate come tali attraverso le misurazioni del questionario stesso. In questo modo è stato possibile evidenziare quali sono le caratteristiche che maggiormente indicano la possibilità che un paziente sia iperteso oppure no, definendo quindi la classe di persone che con più probabilità sono malate ma non stanno ricevendo cure perché non sanno di esserlo. Si è affrontato quindi un problema di classificazione, le cui tecniche di risoluzione sono materia di studio del "*Machine Learning*". In realtà, l'elaborazione di grandi quantità di dati con il fine di apprendere determinate informazioni è di interesse anche per il campo dell'IoT, perché non si limita solamente a studiare come far produrre e conseguentemente raccogliere i dati dei dispositivi *smart*. Il secondo progetto ha riguardato

invece lo sviluppo di un *chatbot* disponibile sulla piattaforma di messaggistica istantanea "Telegram" ad uso dei malati di ipertensione. Questo assistente virtuale deve quindi aiutare il paziente nella propria terapia. Gli permette infatti di registrare le misurazioni di pressione che settimanalmente deve effettuare e gli ricorda di farlo quando passa troppo tempo dall'ultima misurazione. Il sistema permette inoltre di visualizzare medie e grafici delle misurazioni che sono state raccolte cosicché il paziente possa vedere con i propri occhi qual è veramente il suo stato di salute. In realtà questo è fatto anche a beneficio del medico, che può affidarsi ad uno strumento più evoluto del semplice libretto diario in cui il paziente annota tutte le misurazioni. Egli può così vedere con più facilità l'andamento dei valori pressori del paziente. In questo caso l'applicazione dell'intelligenza artificiale è evidente, dato che per comprendere che cosa l'utente vuole che il sistema faccia occorre mettere in pratica delle tecniche di "Natural Language Understanding". È interessante notare come si utilizzi il proprio *smartphone* come interfaccia di primaria interazione, ma questo non significa che non possa diventare una *Thing* capace di raccogliere ulteriori dati.

Capitolo 1

IoT e AI nel mondo dell'healthcare

L'*Internet of Things* e l'*Artificial Intelligence* sono argomenti che non avrebbero bisogno di presentazioni. Ormai sulla bocca di tutti, quelle che ora sono concrete realtà, che possiamo toccare con mano, solo qualche decina di anni fa sembravano pura fantascienza. Nel 1957 il sogno di Herbert Simon, creatore del "General Problem Solver", uno dei primi *software* capaci, seppur rudimentalmente, di imitare il ragionamento umano, era di vedere da lì a dieci anni il primo calcolatore capace di battere un campione di scacchi [3]. Come tante predizioni troppo entusiaste, non si realizzò, se non quarant'anni più tardi grazie al progetto "Deep Blue" di IBM. Similmente, nel 1964 lo scrittore di fantascienza Arthur C. Clarke immaginava una futura rete di calcolatori così potente da "permettere ad un neurochirurgo di operare ad Edimburgo un paziente in Nuova Zelanda" [2]. Anche in questo caso, la predizione non si è avverata, o almeno non ancora, ma quello che è certo è che Kevin Ashton nel 1999 ha posato la prima pietra per questa rete che verrà, introducendo al mondo la nozione di IoT [1]. In ogni caso, vogliamo comunque prodigarci in una breve introduzione di questi due concetti, così da poter meglio capire con che cosa avremo a che fare.

1.1 Artificial Intelligence

Banalmente, si può definire l'intelligenza artificiale come la capacità di un calcolatore di esibire un comportamento intelligente, così come mostrato dagli esseri umani. Questa definizione è però un po' vaga, se non direttamente troppo astratta, per poterne ricavare una materia di studio. Un problema che questa definizione ha, tra tutti gli altri, è quello di capire come essere sicuri che il comportamento della macchina sia effettivamente intelligente, eviden-

ziato già dal filosofo John Searle nel 1980 [14]. Usando una definizione più rigorosa, l'intelligenza artificiale è lo studio "dell'abilità di un sistema di interpretare correttamente informazioni esterne, imparare da queste informazioni e usare le nozioni apprese per raggiungere obiettivi e completare compiti specifici attraverso un adattamento flessibile" [6]. Questa definizione è già più interessante perché permette di far emergere parte di quelli che sono gli obiettivi o i problemi che da sempre questo campo di studi si pone di raggiungere o di risolvere. Il primo che potrebbe venire in mente è quello di costruire sistemi capaci di "ragionare", o meglio di risolvere i problemi, proprio come fanno gli esseri umani, nell'ottica di mettere in pratica le nozioni apprese dai dati interpretati, come appunto detto. Questo però pone un obiettivo ancora più a monte, ovvero come rappresentare la conoscenza di un certo dominio applicativo in maniera tale che possa essere "digeribile" da un calcolatore, le cui metodologie di elaborazione dei dati sono ben diverse da quelle degli esseri umani. Dopodiché, ammettendo di essere riusciti nel precedente obiettivo, si pone il problema di trovare il modo di far apprendere ad una macchina le giuste nozioni assieme al problema di come mettere in relazione tutte quelle che nel tempo il sistema è stato in grado di accumulare. Si vuole infatti che un calcolatore acquisisca sempre più conoscenza su di un dato argomento, anziché "confondersi". Una volta raggiunta una conoscenza sufficiente sul dominio applicativo di interesse, per poter portare a termine i propri compiti, è per il sistema necessario capire come pianificare autonomamente le proprie azioni in modo da poter raggiungere il proprio obiettivo. Obiettivo che potrebbe essere deciso della macchina stessa oppure indicato esternamente. In questa introduzione seguono poi tutta una serie di scopi dell'intelligenza artificiale che hanno a che fare con il modo con il quale una AI può interagire con il mondo, cioè le diverse modalità con cui può concretamente mettere in atto la conoscenza appresa. Si possono infatti costruire AI che hanno lo scopo di interpretare il linguaggio umano e rispondere così come se fosse una persona vera a farlo, di muoversi o di manipolare oggetti autonomamente, di "percepire" il mondo come un essere umano, cioè di essere capace di interpretare che cosa sta succedendo attorno a sè attraverso l'uso dei sensori di cui è dotato, oppure di interagire direttamente con gli altri esseri umani dando l'illusione di essere loro pari.

1.2 Machine Learning

Quando si parla di *Machine Learning* taluni lo identificano come parte della più ampia materia della *Artificial Intelligence*, essendo nato come branca di questa stessa. Altri sostengono che, utilizzando approcci di tipo statisti-

co e probabilistico non consoni a quello che è il cuore del campo di ricerca dell'AI, il *Machine Learning* sia ad oggi una disciplina totalmente differente dall'intelligenza artificiale e goda di uno status autonomo [5]. Quale che sia la tassonomia tra queste discipline, è evidente la connessione che esiste tra le due. Il ML si occupa di studiare algoritmi che migliorano il proprio comportamento attraverso l'esperienza [11], ovverosia attraverso la sempre maggiore quantità di dati a cui hanno accesso. Quindi, alla base di entrambi i campi di studio, si trovano sistemi che riescono ad "apprendere", ovvero a modificare in positivo le proprie azioni per raggiungere un fine ultimo determinato esternamente o autonomamente. Quello che caratterizza nello specifico gli algoritmi di ML, al di là della struttura dell'algoritmo di apprendimento, che sia la rete neurale del *Deep Learning* o il modello lineare dello *Statistical Learning*, è che all'interno di una specifica famiglia di modelli matematici cercano quello che meglio è capace di descrivere l'andamento di un certo insieme di dati. Il loro scopo è perciò quello di individuare le correlazioni presenti all'interno di un insieme di dati e quindi di generalizzare la conoscenza appresa su un insieme di dati più ampio dell'originale. Il modello migliore è determinato come quello capace di minimizzare il valore di una determinata funzione, detta funzione obiettivo o funzione di errore. Come si può notare, la sua definizione è meno farraginosa ma allo stesso tempo meno ampia di quella dell'intelligenza artificiale. Questo però non significa che le applicazioni di questa disciplina siano a loro volta meno ampie, anzi.

1.3 Healthcare

L'aiuto che l'intelligenza artificiale può dare nel campo della medicina è enorme e per dimostrarlo si vuole portare l'esempio dell'AI "Watson". "Watson" è stato sviluppato da IBM con fini totalmente dimostrativi nel 2011 per battere i campioni di un noto *game show* televisivo americano, "Jeopardy!". Il lavoro fatto da IBM è stato talmente sorprendente che già nel 2013 è stata realizzata la prima applicazione commerciale di Watson: un sistema a supporto delle decisioni dei medici nel trattamento dei pazienti malati di carcinoma polmonare presso il centro "Memorial Sloan-Kettering Cancer Center" di New York. Da lì in poi, le applicazioni di Watson in campo medico sono state innumerevoli. Occorre dire che non sono stati tutti casi fortunati, dato che non tutti i numerosi progetti che IBM ha perseguito sono andati a buon fine. Questo è stato causato anche dalle modalità di lavoro proprie del medico, che non si accordano facilmente con le capacità del sistema. Questo però non vuol dire che non siano stati fatti dei progressi, in questo senso è particolarmente interessante il progetto "Watson for Genomics". Esso ha avuto come risultato

un sistema di AI capace di restituire, data la mappa delle mutazioni genetiche nel DNA di un paziente, gli studi e le terapie farmacologiche rilevanti per esse, suggerendo quindi possibili tumori che la persona può sviluppare e conseguentemente cure oncologiche note per essi. Nel 32% dei pazienti è riuscito ad evidenziare mutazioni geniche rilevanti non individuate da commissioni mediche costituite allo scopo nel giro di tre minuti o meno [15].

Per quanto riguarda il *Machine Learning*, gli esempi di applicazione nel campo della medicina sono innumerevoli, molti dei quali appartenenti alla "Computer Aided Diagnosis" o CADx. Questo ramo ibrido tra medicina e *Machine Learning* è volto, come dice il nome, al supporto nell'analisi delle immagini mediche, come radiologie, tomografie, scansioni ad ultrasuoni e simili. Il primo sistema ad utilizzare queste tecniche ufficialmente approvato dalla "Food and Drug Administration" degli Stati Uniti d'America, denominato "ImageChecker", è stato realizzato nel lontano 1998, ma molti sono stati i progressi fatti da allora. "ImageChecker" all'epoca si occupava di analizzare le mammografie nella ricerca di possibili tumori al seno, ed il ML è tutt'ora utilizzato in questo tipo di analisi, ma oggi le sue applicazioni si hanno anche nell'analisi dei risultati delle Tomografie Computerizzate per l'individuazione del carcinoma bronchiale, nell'analisi delle immagini derivanti da angiografie coronariche oppure ancora nell'analisi delle auscultazioni per la rilevazione di difetti cardiaci congeniti. Non solo, in alcuni casi i risultati dell'applicazione del *Machine Learning* nell'ambito dell'analisi di immagini mediche sono stati addirittura sorprendenti, secondo un articolo apparso l'anno scorso su Nature Medicine [16]. In questo articolo sono stati raccolti svariati studi che comparano i risultati ottenuti da reti neurali profonde in alcune specifiche tipologie di analisi con quelli di medici esperti in quel compito. In molti casi, le reti neurali sono riuscite ad eguagliare il lavoro dei singoli medici. Si cita come esempio l'analisi di 64.000 elettrocardiogrammi ottenuti da 29.000 pazienti per la ricerca di fenomeni di aritmia cardiaca, dove una rete neurale profonda è riuscita a eguagliare la precisione di ben sei cardiologi dividendo i pazienti in 14 classi distinte di aritmia.

Un ultimo esempio interessante di AI applicato all'*healthcare*, che permette di introdurre già il legame che questa ha con l'*Internet of Things*, è l'applicazione per la rilevazione della fibrillazione atriale presente in tutti gli Apple Watch. Si tratta di un'applicazione che sfrutta le reti neurali profonde per analizzare il battito cardiaco di una persona alla ricerca di irregolarità riconducibili alla fibrillazione atriale utilizzando i sensori presenti sullo *smartwatch*. L'applicazione ha ricevuto l'approvazione dalla "Food and Drugs Administration" nel 2018 e da quel momento è stata rilasciata su tutti gli orologi *smart* di Apple. In questo caso quindi ci troviamo di fronte ad un classico sistema IoT, il quale vede al centro lo *smartwatch* come controllore dei sensori. Il sistema

registra i dati di questi ultimi per poterli dare in mano alla rete neurale che si preoccupa di effettuare le sue predizioni sulle condizioni di salute di chi indossa l'orologio. Ecco quindi un possibile contatto tra AI e IoT.

1.4 Internet of Things

L'*Internet of Things* invece è un concetto molto distante dai precedenti, ma le cui ramificazioni si estendono fino ad intersecare i precedenti campi di studio, come vedremo. Per utilizzare la definizione che Peter Waher include nella prefazione del suo libro "Learning Internet of Things" [18], "l'IoT è ciò che si ottiene quando si connettono le *Things*, le quali non sono operate da esseri umani, ad Internet". Questa definizione fa emergere tre concetti principali: la non intromissione umana, le *Things* e la connessione delle stesse attraverso Internet. Per quanto riguarda l'indipendenza dalla presenza umana, l'importanza di questo principio era già stata notata ben prima dal già citato Kevin Ashton quando per la prima volta introduceva il termine "IoT". Egli infatti descriveva gli esseri umani, per un sistema informatico, come dei generatori di input lenti, inefficienti, inclini all'errore e che quindi pongono dei limiti consistenti alla quantità e alla qualità delle informazioni disponibili. L'idea che sta dietro ai sistemi IoT è quindi quella di saltare gli intermediari umani e di accedere ai dati che sono prodotti da, o che comunque misurano, eventi o proprietà del mondo reale. L'altro elemento di base su cui vale la pena concentrarci sono le "Cose". Sembra assurdo chiedersi "cosa sia una Cosa", ma in realtà come spesso accade l'uso di un termine molto vago indica la forte malleabilità della sua definizione. In questo senso, una "Cosa" può essere davvero qualsiasi oggetto usato nella vita quotidiana, purché su di esso siano stati applicati sensori o attuatori e possa quindi ricevere o inviare dati. David Rose li definisce "*enchanted objects*" nel suo omonimo libro [13], cioè oggetti banali sui quali è come se fosse stato apposto un incantesimo e, pur sembrandolo, non sono più gli stessi oggetti di prima, ma hanno maggiori e più diverse capacità. Un altro modo per capire che cos'è l'IoT è osservare quali tecnologie sono coinvolte nello sviluppo di sistemi che lo sfruttano. Per questo ci viene in aiuto il *framework stack* introdotto da Timothy Chou nel suo libro "Precision - Principles, Practices and Solutions for the Internet of Things" [4]. Partendo dal livello più basso, troviamo le già citate *Things* e poi, subito sopra, il livello "Connect" che contiene tutte quelle tecnologie che si preoccupano di connettere le "Cose" tra di loro. Ancora sopra troviamo il livello "Collect" che contiene invece tutte le tecnologie che hanno il compito di raccogliere i dati, anche in grandi quantità, che sono stati prodotti dalle *Things*. Ma i livelli più interessanti sono i due più in alto, ovverosia "Learn" e

"Do". Nello *Internet of Things* non ci si accontenta semplicemente di costruire una rete di oggetti intelligenti, ma lo si fa con uno scopo ben preciso: quello di poter costruire modelli che analizzano le grandi quantità di dati raccolti, apprendono da esse e poi mettono in pratica la conoscenza ottenuta. Appare ora esplicito il collegamento con l'intelligenza artificiale e nello specifico con il *Machine Learning*: questi studiano tecnologie le quali occupano i livelli più alti dello *stack* tecnologico dello *Internet of Things*, il quale perciò ne dipende fortemente.

L'IoT oggi è ovunque, persino nelle nostre case, perciò anche in questo caso la quantità di applicazioni che si possono portare in esempio è enorme, ma dato l'argomento di questa tesi ci focalizzeremo su alcuni esempi in campo *healthcare*. Il più semplice ed immediato è l'oggetto che molti di noi portano già al polso già da alcuni anni: lo *smartwatch*. È chiaro come rappresenti un *enchanted object*: è passato dalla semplice funzione di segnare l'ora, o anche i giorni e i mesi in alcune varianti più avanzate, a poter telefonare, inviare messaggi, navigare in rete, eccetera. Tutti gli *smartwatch* in commercio, dallo "Apple Watch" di Apple al "Galaxy Watch" di Samsung, sono dotati di svariati sensori a bordo, pensati per il fitness e per il monitoraggio dello stato di salute di chi li indossa. Recentemente "Omron", una delle società leader nel settore della produzione di sfigmomanometri, ha lanciato uno dei primi rilevatori di pressione sanguigna *wearable*, cioè che si può portare al polso come se fosse un orologio. È chiaramente molto più che un semplice orologio, dato che oltre alle funzioni di quest'ultimo ha la possibilità di monitorare parametri vitali come pressione minima, pressione massima, frequenza cardiaca, numero di passi fatti, ma anche di poter travasare i dati raccolti in un'applicazione per cellulare e tracciare dei grafici che mostrino l'andamento degli stessi. Questo dimostra come l'IoT permette e permetterà sempre di più di sviluppare sistemi capaci di supportare la salute della persona a tutto tondo, monitorando continuamente i suoi parametri vitali e cercando di avvertire il soggetto di eventuali problematiche. Per di più, questi parametri, una volta registrati, possono essere comunicati al proprio medico per ricevere una migliore e più completa analisi del proprio stato di salute.

1.5 Chatbot

Nati come una delle tante applicazioni della *Artificial Intelligence*, i "chatbot", detti anche assistenti virtuali, sono ormai diventati sistemi degni di una dignità propria. Storicamente, sono stati visti come il Sacro Graal dell'AI, dato che Alan Turing, considerato non a torto uno dei padri dell'informatica, già parlava di macchine capaci di simulare le capacità di dialogo umano nel suo

celebre test [17]. È facile capire come mai questi sistemi abbiamo da sempre attirato l'attenzione e la curiosità degli scienziati. Il loro spiccato approccio conversazionale nell'interazione con l'*end user* rende molto più facile pensarli come "umani" rispetto a qualsiasi altro sistema informatico che si possa immaginare. In ogni caso, è abbastanza evidente perché lo sviluppo di questi sistemi appartenga al campo dell'intelligenza artificiale: perché funzionino è necessario che abbiano un certo grado di abilità nella comprensione del linguaggio naturale, cioè il sistema deve essere capace di effettuare, in un certo qual grado, "Natural Language Understanding". Occorre infatti che il sistema sia allenato ad ottenere informazioni significative da frasi complesse, non necessariamente fornitegli in formato testuale, e possa imparare da esse delle nozioni utili per portare avanti i propri scopi. Questi, ad oggi, possono essere i più disparati, non solo il semplice portare avanti una discussione come se ci si trovasse davanti ad un altro essere umano. Infatti, questi assistenti virtuali sono anche legati a stretto giro con l'*Internet of Things*. Questo perché la loro natura intrinseca è proprio quella di essere degli assistenti, di aiutare chi li interpella. La possibilità di connettersi ad altre *Things* è la naturale estensione di questi sistemi, che permettono all'utente di esprimere un comando e vedere i dispositivi *smart* di cui il *chatbot* è *controller* agire, senza dover utilizzare manualmente alcuna interfaccia composta da levette e bottoni di varia natura.

Di questo ne è testimonianza qualsiasi assistente virtuale presente all'interno di uno *smart* speaker, da "Alexa" di Amazon ad "Assistente Google" di Google. Se non potessero interfacciarsi con i dispositivi intelligenti presenti all'interno della casa, la loro utilità rimarrebbe molto limitata. I *chatbot* sono in realtà di fortissimo interesse per molti settori industriali, dato che permettono di automatizzare una delle parti più delicate della fidelizzazione con il cliente, ovverosia il *customer care*. Anche in questo caso, data la fortissima capacità dei *chatbot* di adattarsi a qualsiasi contesto di utilizzo, esistono dei casi d'uso di successo nel campo dell'*healthcare*. Un esempio molto noto nato di recente è il *chatbot* disponibile per la piattaforma di messaggistica istantanea "Whatsapp" dell'Organizzazione Mondiale della Sanità per fornire aiuto e informazioni al grande pubblico sul Covid-19 e sul SARS-CoV-2. Benché sia stato descritto usare tecniche di ML, ad un primo impatto non è evidente a che livello dello sviluppo esse siano state impiegate. In ogni caso, sicuramente è un esempio che mostra il potenziale che questi sistemi hanno nel campo della medicina.

Capitolo 2

Il problema dell'ipertensione

Secondo lo "Annuario statistico italiano" che l'Istat ha pubblicato nel 2019 [10], l'ipertensione è la malattia cronica più diffusa nel nostro Paese, con il 18,1% degli italiani intervistati che ne soffre. È perciò di fondamentale importanza trovare le migliori soluzioni che sono in grado di contrastarla. Esistono svariati modi per farlo, uno di questi è fare azioni di prevenzione della malattia. Questo significa educare le persone ai comportamenti corretti da seguire per minimizzare la probabilità di svilupparla, soprattutto coloro che sono più a rischio. Un altro modo è invece quello di agire più direttamente, fornendo migliori cure o migliorando quelle esistenti e fornendole in maniera più tempestiva, cercando di raggiungere il più presto possibile anche coloro che non sanno di soffrirne. Ma come si può identificare chi è più a rischio? Come si può migliorare la terapia ipertensiva che è già correntemente somministrata ai pazienti? In questa tesi si è cercato di trovare soluzioni a questi due problemi.

2.1 Prevenzione

Come per qualsiasi malattia, la prevenzione è importante tanto quanto la cura. Individuare per tempo la presenza dell'ipertensione in un paziente, quando i sintomi sono ancora poco presenti o non sono presenti affatto, potrebbe portare ad un trattamento più efficace della malattia stessa, migliorando le possibilità che il paziente ne rimanga debilitato il meno possibile. In questo contesto si inserisce la ridefinizione delle linee guida per il trattamento dell'ipertensione, che viene fatta periodicamente. Essa è volta a prendere atto dei cambiamenti che intercorrono nei rapporti tra la malattia e la popolazione, delle nuove scoperte che sono emerse dagli studi più recenti e così via. Il tutto con il fine ultimo di riuscire ad elaborare un protocollo medico per la cura dell'ipertensione sempre più efficace. Uno tra i compiti di queste linee guida è definire la parte della popolazione che è più a rischio nel soffrire di

ipertensione, in modo tale da far capire a chi se ne occupa su chi concentrare maggiormente i propri sforzi nel cercare di individuare nuovi pazienti ipertesi che non sapevano di esserlo. Sarebbe utopistico pensare che lo Stato possa avere sufficienti risorse per poter effettuare uno *screening* contro l'ipertensione a tutti i cittadini italiani periodicamente. Peraltro, sarebbe anche uno sperpero delle risorse pubbliche se questo significasse effettuarlo anche su quei sottogruppi della popolazione italiana che sono noti non presentare casi di ipertensione se non rarissimamente. Proprio per questo motivo è molto importante identificare le caratteristiche di quella frazione della popolazione che con sufficiente probabilità può stare soffrendo di ipertensione. In questo modo diventa più facile individuare persone ipertese che non sapevano di esserlo e svolgere uno screening più efficace. Per di più, non è nemmeno possibile affidarsi ad un'autodiagnosi fatta delle singole persone: molti potrebbero non presentare sintomi, o presentarli in forma talmente lieve da poterci convivere e non allertare mai nessuno specialista, di fatto trascurando in maniera colpevole la propria salute. Come detto in precedenza, anche se in un diverso contesto, la componente di raccolta dei dati più soggetta ad errori è proprio quella umana, perciò è importante che sia lo screening ad andare dal paziente e non viceversa.

2.2 Cura

Quanto detto fino adesso non significa che la cura dell'ipertensione non sia altrettanto importante, anzi. Proprio a causa della cronicità di questa malattia, del non risolversi mai dei suoi sintomi, è assolutamente importante curare il paziente al meglio, affinché possa convivere con la propria malattia nel modo meno debilitante possibile e possa mantenere un buono stato di salute. Sempre a causa del suo decorso senza termine, unito a dei sintomi non particolarmente gravi nella stragrande maggioranza dei casi, molti malati trascurano la loro ipertensione. È facile capire come, sulla base di quanto appena detto, una grossa fetta della terapia per la cura di questa malattia è sulle spalle del paziente stesso, che dovrebbe monitorarla attentamente sotto raccomandazione del medico ed avvertirlo nel momento nel quale la cura sta perdendo di efficacia. La misurazione dei valori di pressione sanguigna e di frequenza cardiaca andrebbe effettuata qualche volta a settimana, con un'intensificazione ad una volta al giorno poco prima del *check-up* medico annuale per il controllo della condizione di salute del paziente. Molte persone invece finiscono per misurarla qualche volta al mese o addirittura solo qualche volta all'anno, rimanendo nel buio per quanto riguarda la propria salute e lasciando nel buio anche il medico che dovrebbe aiutarli. Inoltre, il medico si basa proprio sulla valutazione del-

le misurazioni effettuate nell'ultimo anno per decidere se la terapia prescritta ad un paziente è adeguata o meno. Senza queste misurazioni non è possibile stimarne la correttezza ed è per questo che esistono pazienti che continuano a prendere gli stessi farmaci con le stesse dosi per anni senza poter sapere se effettivamente la terapia è giusta per loro oppure no. C'è chi, addirittura, dimentica anche di prendere la terapia prescritta dal medico, vanificando ogni possibile sforzo di cura della sua ipertensione. Da ultimo, è importante per il medico assicurarsi anche che le misurazioni di pressione siano eseguite correttamente. Questo perché sono noti alcuni fattori che possono alterare i valori di pressione di un paziente, ad esempio: il fumo, le forti emozioni, i pasti sono capaci di aumentare i valori di pressione sanguigna, mentre misurarsi la pressione di sera, di notte oppure la mattina presto fa in modo che i valori siano più bassi. È perciò importante che il paziente sia a conoscenza di ciò perché le misurazioni di pressione non siano falsate e che il medico sia a conoscenza di tutte le informazioni utili possibili sulle misurazioni per capire se sono state falsate o meno.

2.3 Soluzioni

Questi problemi ci sono stati portati dal dottor Martino Pengo, che lavora presso lo "Istituto Auxologico Italiano" di Milano. Egli fa parte di una commissione di medici per la ridefinizione delle linee guida per la prevenzione e la cura dei malati di ipertensione e lavora a stretto contatto con i propri pazienti che soffrono di questa malattia. Per cercare di risolvere il primo dei problemi elencati nelle precedenti sezioni, ci ha fornito un *dataset* ottenuto elaborando le risposte ai questionari che sono stati raccolti durante la "Giornata Mondiale dell'Ipertensione" nelle edizioni degli ultimi cinque anni. Questi questionari permettevano di suddividere le persone che li hanno compilati tra ipertesi noti e non e questi ultimi a sua volta in probabili nuovi ipertesi e non ipertesi, grazie a delle misurazioni di pressione arteriosa e di frequenza cardiaca che sono state raccolte assieme al questionario stesso. In questo modo è stato possibile portare avanti un'analisi dei dati tramite gli strumenti del *Machine Learning*, per poter individuare quali sono le caratteristiche che indicano con maggior probabilità che una persona può stare soffrendo di ipertensione senza saperlo. Individuare queste specifiche caratteristiche, che possono variare da semplici dati antropometrici come età o peso alle patologie di cui una persona soffre, ha quindi poi come risultato quello di poter applicare questa conoscenza alla ricerca del sottogruppo della popolazione italiana che è più incline a sviluppare ipertensione in generale, facendoci raggiungere lo scopo prefissato.

Per poter risolvere invece il secondo dei problemi già evidenziati, il dottor Pengo ci ha chiesto di sviluppare un sistema capace di aiutare il paziente nella cura della sua malattia. Il sistema dovrà quindi essere capace di ricordare al paziente di misurarsi la pressione e di prendere la propria terapia periodicamente. Dovrà però farlo con una frequenza adeguata, ciò significa nè in maniera troppo poco frequente, che farebbe perdere di utilità al sistema, nè in maniera troppo frequente, che lo renderebbe pedante e scoraggerebbe il suo utilizzo. Dovrà permettere anche di rendere il medico partecipe delle misurazioni che il sistema sta raccogliendo, ponendosi come alternativa al libretto diario che il paziente dovrebbe tenere e che il medico una volta all'anno deve visionare. In più, a beneficio del medico il cui lavoro è quindi semplificato, ma anche del paziente che ottiene più informazioni sul suo stato di salute, si vuole che il sistema mostri dei riepiloghi sui dati che ha acquisito recentemente. Infine, il sistema dovrà anche fornire degli utili consigli al paziente affinché le misurazioni possano essere quanto più veritiere possibili e non falsate in positivo o in negativo da fattori esterni.

Capitolo 3

Analisi dei dati

In questo capitolo si discuterà della soluzione data al primo dei due problemi evidenziati precedentemente, quello di analisi dei dati. La soluzione è stata costruita tramite un Notebook Jupyter per poter vedere immediatamente i risultati del codice che veniva mano a mano scritto. Inoltre, il linguaggio Python è stato scelto per poter utilizzare gli strumenti forniti dalla più celebre e più completa libreria per il *Machine Learning* attualmente in uso: "scikit-learn". Questa ha poi portato con sè il fatto di poter utilizzare una serie di ulteriori librerie che normalmente le fanno da contorno: "pandas" per l'elaborazione di dati strutturati tramite la sua struttura dati "DataFrame", "matplotlib" e "seaborn" per il tracciamento di grafici e "numpy" per la manipolazione di vettori numerici.

3.1 Introduzione al problema e comprensione delle variabili

Il problema richiedeva di predire se una persona adulta potesse essere potenzialmente ipertesa o meno sulla base di un certo insieme di parametri antropometrici forniti. È perciò evidente che ci si trova davanti ad un problema di classificazione binaria, in quanto è necessario suddividere i pazienti in due classi: "potenzialmente ipertesi" e "non ipertesi". La limitazione dei pazienti considerati ai soli adulti si è resa necessaria perché l'ipertensione viene individuata secondo modalità differenti negli adulti e nei bambini e perciò non è possibile portare avanti la stessa analisi su questi due gruppi distinti di persone. Per di più, adulti e bambini presentano dati antropometrici molto differenti tra di loro e nello stesso gruppo delle persone minorenni la correttezza dei parametri stessi varia di anno in anno. In conclusione, per rendere più veritieri i risultati ottenuti, sono stati esclusi tutti i minori di 18 anni dal

problema. Tutti i dati, rigorosamente resi anonimi, provengono da questionari compilati da persone intervistate nell'ambito della "Giornata Mondiale dell'Ipertensione", promossa dalla "Società Italiana dell'Ipertensione Arteriosa", e contengono più di 37 mila istanze. I dati contenevano la sede e l'anno in cui si è svolta la raccolta dei questionari, è possibile perciò dedurre che sono stati raccolti da cinque anni fa all'anno scorso. Oltre a queste informazioni erano presenti tre misurazioni di pressione diastolica, di pressione sistolica e di frequenza cardiaca ed assieme a queste i dati antropometrici della persona a cui erano state fatte le misurazioni, ovvero età, peso in chilogrammi, altezza in centimetri e sesso biologico. Infine, seguivano le domande del questionario vere e proprie. La prima domanda, con risposte non mutualmente esclusive, verteva sulle condizioni di salute del paziente, chiedendogli se fosse vero o falso che:

- fumasse
- soffrisse di insufficienza renale
- fosse diabetico
- avesse sofferto di eventi cardiaci ischemici in passato, come ad esempio infarto, sindrome coronarica acuta, angioplastica, bypass, eccetera
- avesse il colesterolo alto
- avesse sofferto di eventi cerebrali in passato, come ad esempio ictus
- almeno uno dei familiari dell'intervistato soffrisse o avesse sofferto in passato di eventi cardiaci ischemici

Qualora l'ultima risposta fosse stata indicata come vera, occorreva poi indicare quale o quali familiari soffrissero di eventi cardiaci ischemici.

La seconda domanda invece riguardava sempre le condizioni di salute del paziente, entrando però nel merito dell'ipertensione. Si chiedeva infatti se il paziente fosse iperteso e prendesse farmaci per contrastare questa malattia, se fosse iperteso e non prendesse farmaci, se non lo fosse oppure se non sapesse di esserlo. Questa domanda ha svolto un ruolo centrale nell'analisi dei dati perché ha permesso di distinguere quali erano le istanze utili per costruire i modelli predittivi e quali no. Infatti, tutti coloro che hanno risposto sì a questa domanda sono stati scartati, dato che non sono possibili ipertesi, ma ipertesi certi, visto che ne sono a conoscenza. In altre parole, sono stati utilizzati durante l'analisi solamente i dati di quelle persone che non avessero scelto le prime due risposte in questa domanda. La terza domanda era l'ultima che

riguardava le condizioni di salute dell'intervistato e gli chiedeva se si fosse mai recato al pronto soccorso per problemi inerenti l'ipertensione.

Dopodiché, seguivano alcune domande di cultura generale sull'ipertensione. Nella quarta domanda si chiedeva infatti se il paziente sapesse in che percentuale nei paesi occidentali si soffrisse di ipertensione. Questa domanda non è stata ritenuta utile ai fini della costruzione di un modello predittivo e per questo motivo le sue risposte sono state semplicemente scartate. Nella quinta domanda, con risposte non mutualmente esclusive, si chiedeva invece se l'intervistato sapesse quali erano i danni provocati dall'ipertensione e venivano indicate come possibili risposte le seguenti:

- ischemia o infarto cardiaco
- ischemia o infarto cerebrale
- insufficienza renale
- insufficienza epatica
- cecità
- diabete mellito

Nella sesta domanda, sempre con risposte non mutualmente esclusive, si chiedeva se l'intervistato sapesse quali sono le buone pratiche da adottare nella vita di tutti i giorni per ridurre l'insorgenza di ipertensione arteriosa e di malattie cardiache in generale. Le risposte presenti erano:

- Seguire una dieta con poche calorie e molte proteine
- Limitare il consumo di alcolici
- Seguire una dieta con pochi grassi e poco sale, ricca di fibre e vitamine
- Bere un bicchiere di vino rosso al giorno
- Fare almeno 30 minuti di attività fisica al giorno
- Fare esclusivamente attività sportiva molto intensa
- Non bere caffè
- Evitare il fumo
- Fare regolari controlli medici, anche se non si hanno disturbi
- Fare un controllo medico appena compaiono dei disturbi, ma non prima

Queste due domande, anche se nello stile della quarta, sono state al contrario considerate. Questo perché le risposte evidenziano un certo tipo di nozioni possedute dall'intervistato che lo possono portare ad intraprendere un differente stile di vita o un diverso modo di nutrirsi, entrambi i quali sicuramente hanno un impatto sull'insorgenza dell'ipertensione in un soggetto a rischio.

La settima domanda invece riguardava la frequenza di misurazione della pressione arteriosa e si chiedeva se venisse effettuata meno di una volta all'anno, qualche volta all'anno, mensilmente o su base giornaliera.

Le ultime tre domande invece riguardavano il sonno dell'intervistato e chiedevano rispettivamente se esibisse sonnolenza durante il giorno, se russasse mentre dorme e se soffrisse di pause del respiro durante il sonno. Queste ultime due domande ammettevano anche la possibilità che la persona non sapesse se si applicavano nel suo caso oppure no.

3.2 Analisi esplorativa

Una volta compreso qual era il significato dei dati, ci si è dedicati all'analisi delle singole variabili per osservare le loro distribuzioni e gli *outlier* all'interno delle stesse. Si è innanzitutto analizzato il sesso di appartenenza degli intervistati, escludendo i valori lì dove non è stata possibile interpretare la risposta. Come si può notare dalla figura 3.1, il valore della variabile è stato equilibrato, con una lieve sproporzione verso le donne, probabilmente perché mediamente in tutte le fasce d'età sono presenti più donne che uomini. Analizzando la variabile inerente all'età degli intervistati si può notare dalla figura 3.2 che

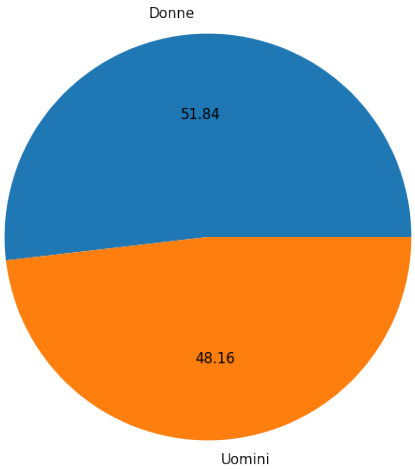

Figura 3.1: Percentuali sul sesso degli intervistati nel questionario

non è presente nessun tipo di *outlier*.

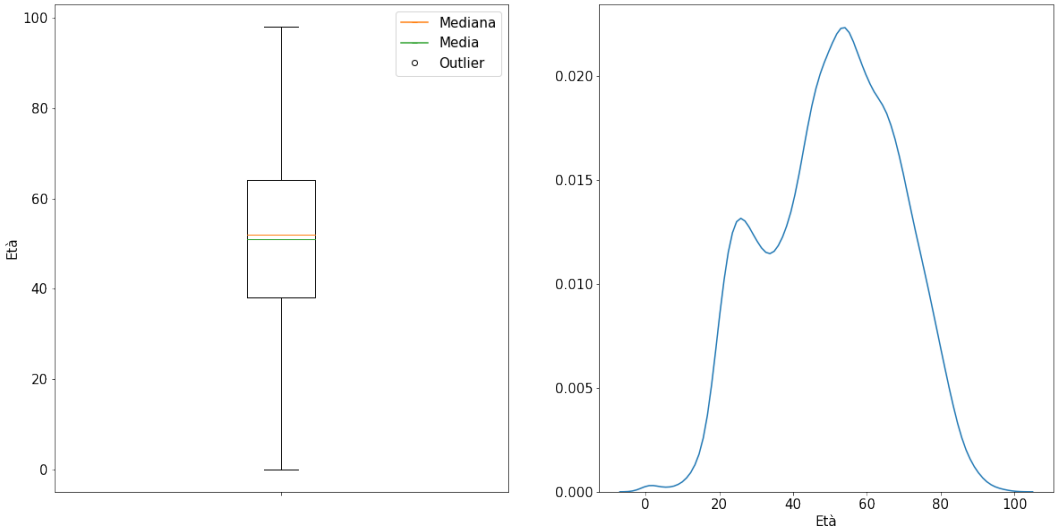

Figura 3.2: Istogramma e stima *kernel* di densità per la variabile relativa all'età degli intervistati

L'analisi di tutte le variabili numeriche successive è stata eseguita allo stesso modo, ovverosia da una parte, tramite un *boxplot*, si è controllato quali e quanti *outlier* erano presenti e dall'altra parte si è costruita la stima *kernel* di densità per i valori della variabile, similmente a quanto fatto per la variabile dell'età precedente. Dopodiché, per verificare che la variabile seguisse una distribuzione normale, si è costruita una curva gaussiana con parametro μ pari alla mediana della distribuzione dei valori e parametro σ pari alla deviazione standard della stessa ed è stata poi sovrimpressa alla stima *kernel* di densità. Questo fatto è sempre stato verificato perché variabili come altezza, peso, pressione sistolica, pressione diastolica e battito cardiaco sono tutte variabili la cui distribuzione può essere generalmente approssimata con una normale e questo è tanto più vero quante più misurazioni abbiamo disponibili, per il teorema centrale del limite. Quindi, con un *dataset* con così tante istanze questo comportamento dei dati è molto evidente, come avremo modo di osservare in seguito durante questa dissertazione. Per non subire l'influenza degli *outlier* in questo processo si è deciso di utilizzare la mediana e non la media come valore per μ, che come si potrà osservare è sempre leggermente sfalsata rispetto alla prima a causa dei punti che non seguono la distribuzione. La verifica del fatto che i dati seguano un andamento gaussiano è stata poi utile in seguito per l'eliminazione degli *outlier*, per poter discernere tra quelli veramente tali e quelli che sono invece deviazioni naturali dal modello matematico. Si anticipa il fatto che i grafici delle variabili di altezza, peso e battito cardiaco non sono

stati costruiti a partire da tutte le istanze del *dataset*. Sono state utilizzate solo una parte di esse per ragioni che saranno più chiare in seguito. Si è quindi applicata l'analisi descritta sulla variabile inerente all'altezza, i cui risultati sono visibili nella figura 3.3.

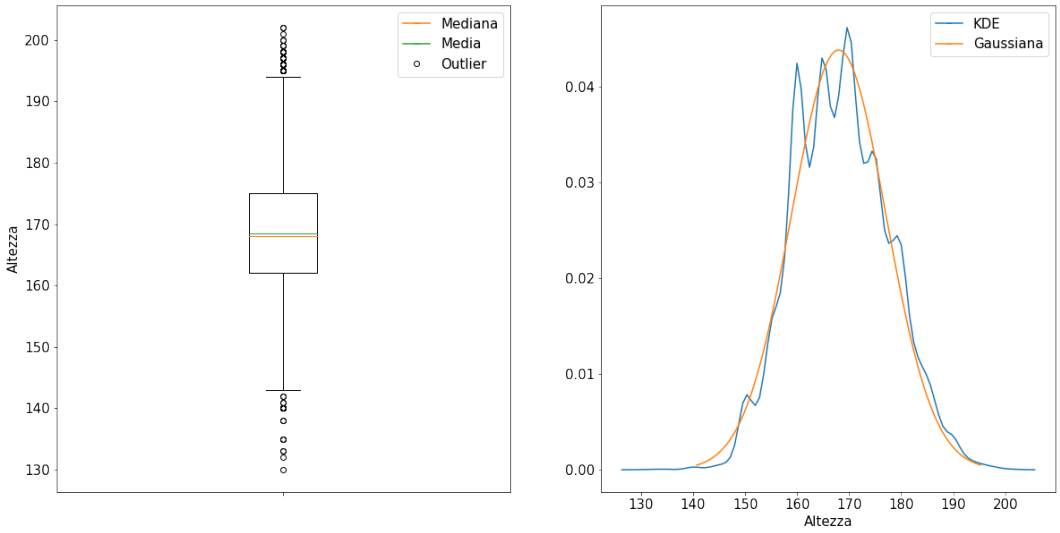

Figura 3.3: Istogramma, stima *kernel* di densità e approssimazione tramite gaussiana della stessa per la variabile relativa all'altezza degli intervistati

Per quanto riguarda il peso invece l'analisi della variabile è visibile nella figura 3.4. Il peso e l'altezza però non si è potuto utilizzarli così, *tout court*. Infatti, come mostra la figura 3.5, esiste una correlazione lineare tra le due variabili. D'altronde è logico che sia così, il peso di una persona è linearmente proporzionale al suo volume, che a sua volta è linearmente proporzionale alla sua altezza. Questo però provocherebbe un fenomeno di multicollinearità che renderebbe instabili i modelli prodotti, il che è fortemente indesiderato. Quindi, come vedremo in seguito, è stata operata una trasformazione del *dataset* che coinvolgesse queste variabili in modo tale che ciò non si verificasse.

Per quanto riguarda le misurazioni di pressione sistolica, diastolica e la frequenza cardiaca, il metodo per la loro analisi è stato, come detto in precedenza, lo stesso che per le altre variabili numeriche. Diversamente da prima però, le tre diverse misurazioni per ciascuno di questi parametri sono state elaborate assieme. Nella figura 3.6 si possono osservare le distribuzioni dei valori delle misurazioni relative alla pressione sistolica, nella figura 3.7 quelle relative alla pressione diastolica e infine nella figura 3.8 quelle relative alla frequenza cardiaca.

Infine, sono state trattate le variabili inerenti alle domande del questionario. Le domande 1, 5 e 6 avevano risposte non mutualmente esclusive e per

questo ciascuna di queste ultime è stata codificata mediante i valori 0 e 1 in ciascuna delle tre domande. Il valore 0 significava che la risposta non era stata selezionata, il valore 1 se invece lo era stata. Nelle altre domande invece, dove le risposte erano mutualmente esclusive, è stato associato ad ogni risposta un valore numerico incrementale a partire da 1. In realtà anche in queste domande

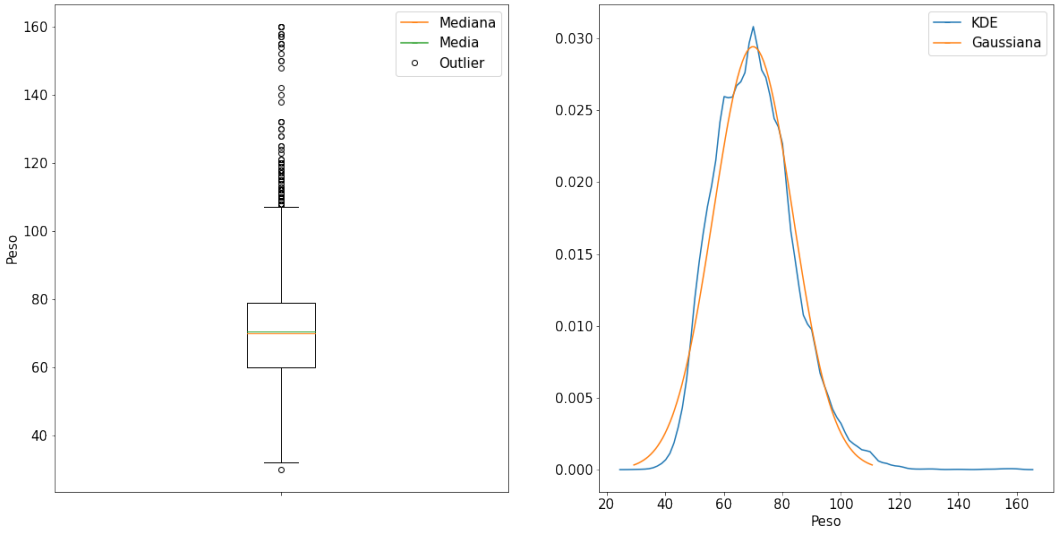

Figura 3.4: Istogramma, stima *kernel* di densità e approssimazione tramite gaussiana della stessa per la variabile relativa al peso degli intervistati

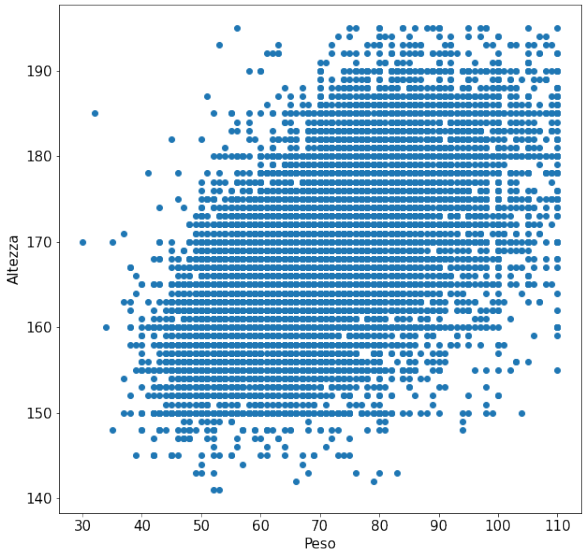

Figura 3.5: Grafico di dispersione che mostra la correlazione tra peso ed altezza delle persone contenute nell'insieme dei dati

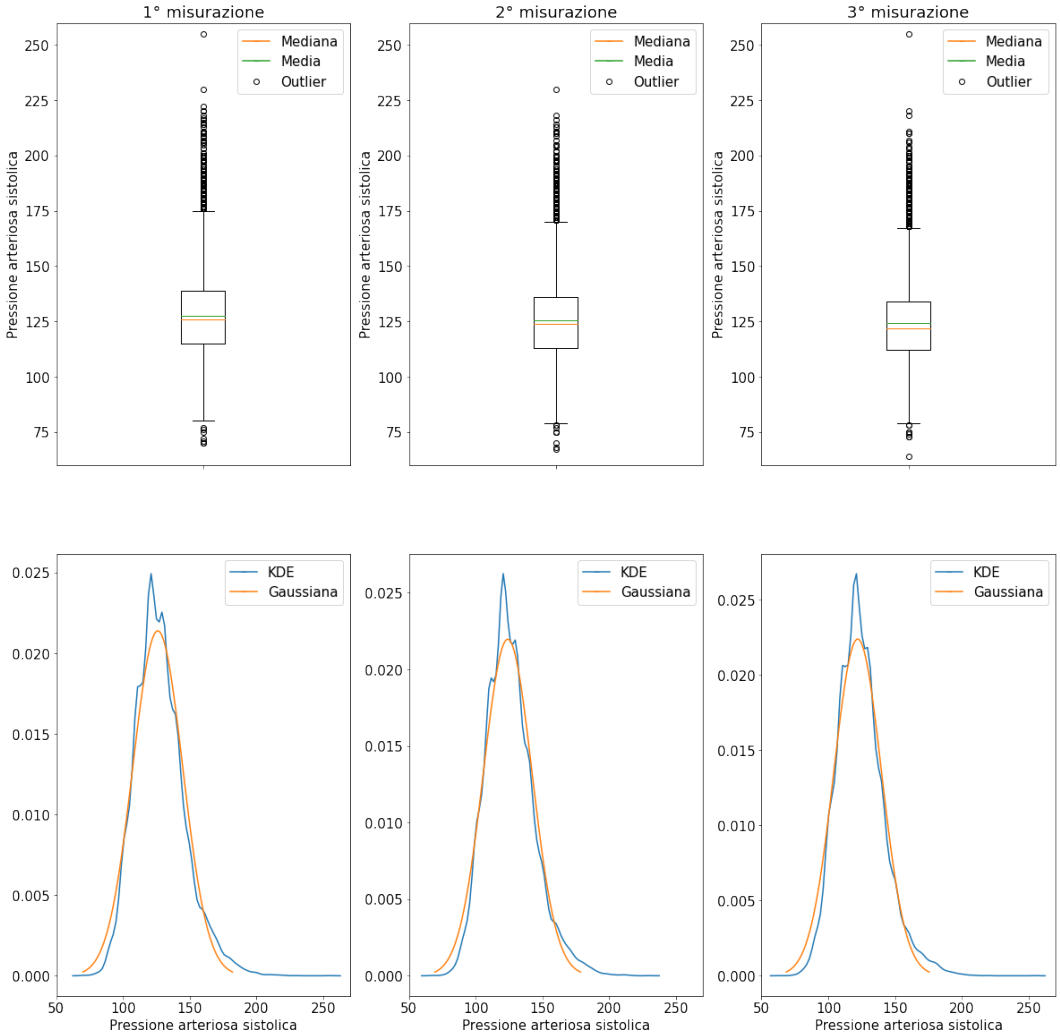

Figura 3.6: Istogramma, stima *kernel* di densità e approssimazione tramite gaussiana della stessa per le variabili relative alle misurazioni di pressione sistolica degli intervistati

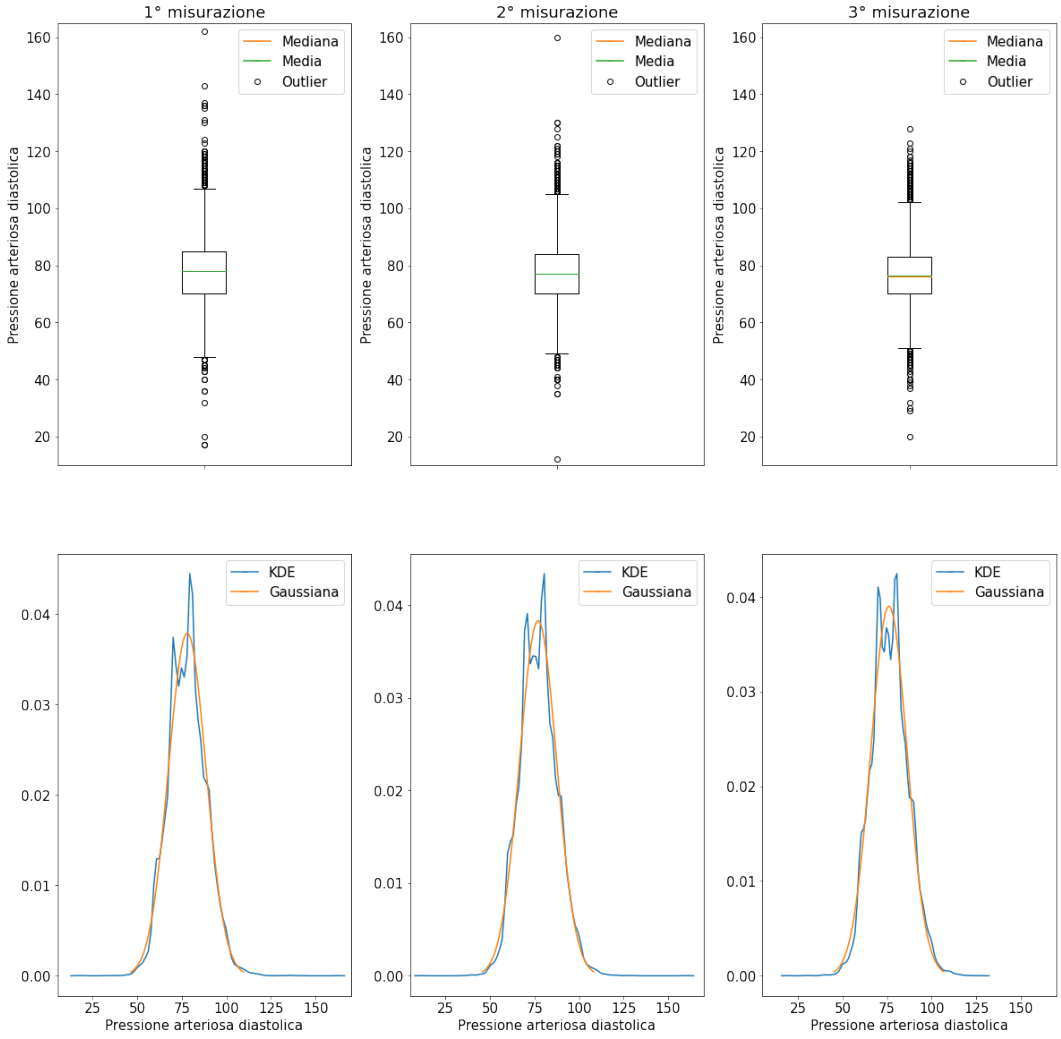

Figura 3.7: Istogramma, stima *kernel* di densità e approssimazione tramite gaussiana della stessa per le variabili relative alle misurazioni di pressione diastolica degli intervistati

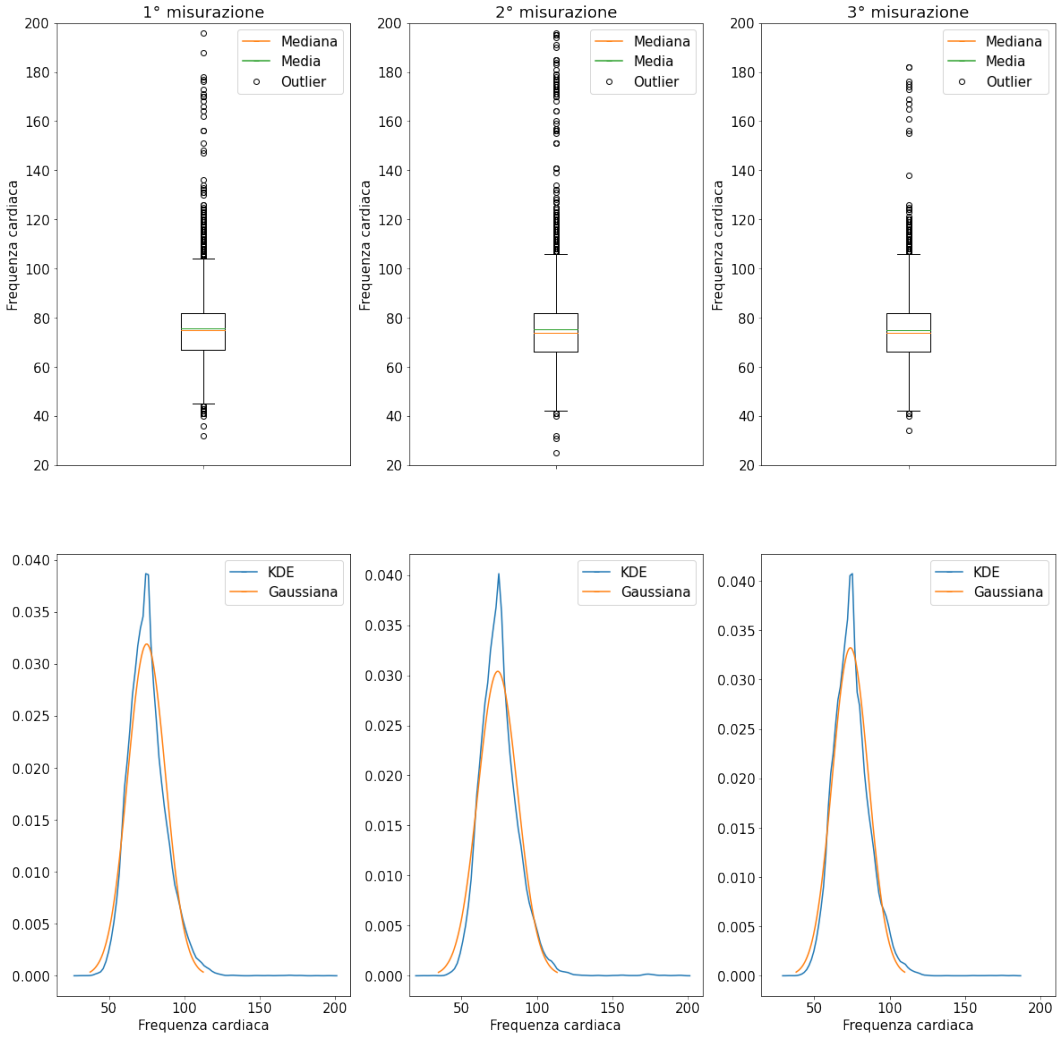

Figura 3.8: Istogramma, stima *kernel* di densità e approssimazione tramite gaussiana della stessa per le variabili relative alla frequenza cardiaca degli intervistati

era presente il valore 0, che non sarebbe dovuto essere ammesso. Sulla base di quanto detto per le domande precedenti, sulla base del fatto che lì dove il valore per il sesso degli intervistati era assente in realtà compariva il valore 0 e sulla base del fatto che il questionario è stato fatto per poter essere letto da un lettore ottico, si è supposto che il valore 0 in queste domande significasse "non scelto". Quindi, quando si è trovato 0, per noi ha significato che l'intervistato non ha annerito nessuna risposta per quella domanda oppure il lettore non è stato capace di interpretare la risposta. È importante sottolineare che nella variabile relativa alla domanda 3 sono stati trovati anche i valori 3 e 5, che non

erano ammissibili sotto nessuna condizione, ma fortunatamente comparivano soltanto in 51 istanze complessivamente.

3.3 Estrazione delle variabili rilevanti

Effettuando un'associazione tra i valori della città in cui si è svolto il sondaggio e la zona d'Italia in cui essa si trova, si è notato, come si può osservare nella figura 3.9 che circa la metà dei sondaggi si è svolto in nord Italia, mentre il restante 50% in maniera all'incirca equa tra centro e sud Italia.

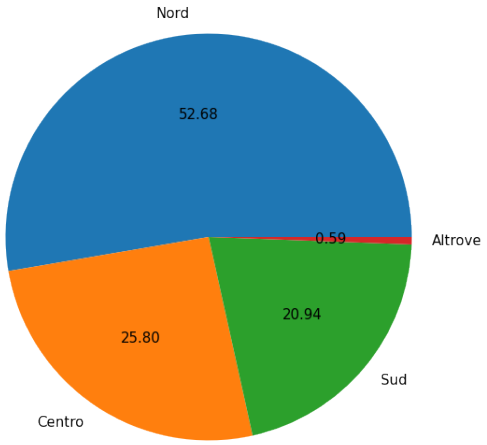

Figura 3.9: Percentuali sulla provenienza dei questionari suddivisa per zone d'Italia

C'è stata una piccola percentuale di persone che è stata intervistata al di fuori dell'Italia, nello specifico in Romania e all'ospedale civile di Misurata, in Libia. La costruzione di questa nuova variabile è stata fatta per sostituire le informazioni fornite da `SEDE`, i cui valori erano molto più frammentati e di difficile interpretazione. Questo è stato fatto attraverso l'uso di un file JSON trasformato poi in "DataFrame" che associava ciascun comune italiano alla regione di appartenenza. In seguito, risaliti alla regione in cui il questionario è stato raccolto, è stato facile costruire un'associazione tra regione e zona d'Italia mediante un semplice *dictionary*, dato che le diverse zone d'Italia sono definite proprio sulla base delle regioni in esse contenute.

Per evitare eventuali fenomeni di multicollinearità tra le variabili `ALTEZZA` e `PESO`, già discussi nella sezione precedente, si è resa necessaria la loro sostituzione con l'indice di massa corporea. Questo indice è calcolato come il

rapporto tra il peso in chilogrammi e il quadrato dell'altezza in metri di un individuo. Per ottenerlo quindi non sono state necessarie ulteriori trasformazioni delle variabili. Una volta costruito il *Body Mass Index*, la sua analisi si è svolta come quella fatta in precedenza per una qualsiasi altra variabile numerica. Essa può essere osservata nella figura 3.10 e ancora una volta notiamo come anche questa variabile segua una distribuzione normale quasi perfettamente.

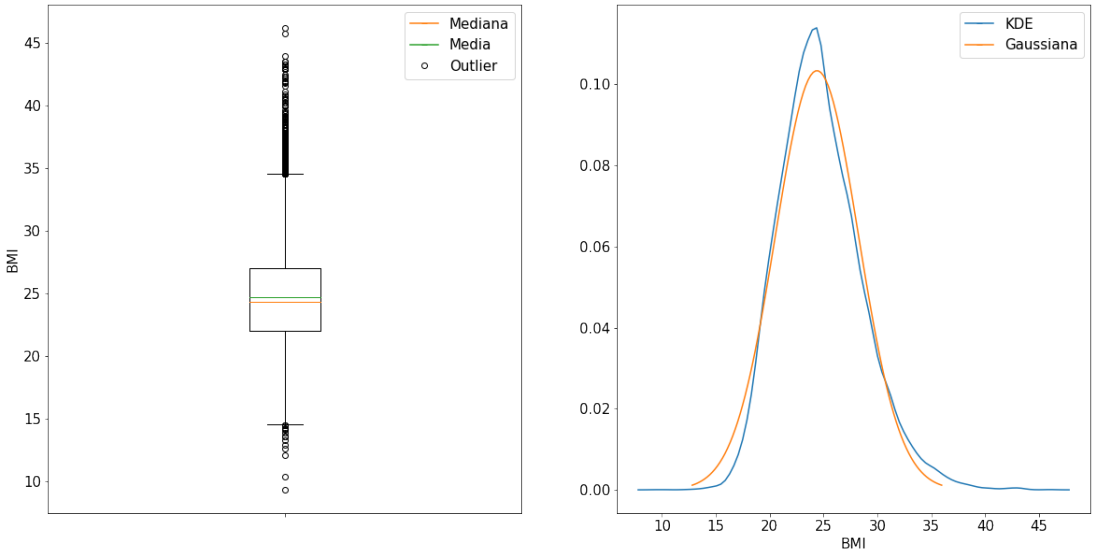

Figura 3.10: Istogramma, stima *kernel* di densità e approssimazione tramite gaussiana della stessa per la variabile relativa al *Body Mass Index* degli intervistati

Come ci si poteva aspettare, le variabili delle misurazioni di pressione sistolica e diastolica sono state utilizzare per costruire la variabile discreta da predire. Il metodo utilizzato ha seguito le linee guida della "American Hearth Association" sull'ipertensione negli adulti [8]. Secondo queste, per poter dire se un paziente è iperteso o meno, occorre che la media delle ultime due rilevazioni effettuate di pressione sistolica risulti superiore a 140 mmHg o la media delle ultime due misurazioni effettuate di pressione diastolica risulti superiore a 90 mmHg. La figura 3.11 mostra le percentuali delle istanze delle due classi così costruite. Come si può notare, è un problema di classificazione sbilanciato, che ha perciò richiesto degli accorgimenti specifici per fare in modo che le soluzioni non degenerassero. Per quanto riguarda invece le misurazioni della frequenza cardiaca, analogamente a quanto fatto per le misurazioni di pressione, è stata costruita una nuova variabile che rappresenta la media delle stesse e questa prende in considerazione solo le ultime due.

Infine, per poter utilizzare i valori della variabile del familiare che ha sofferto

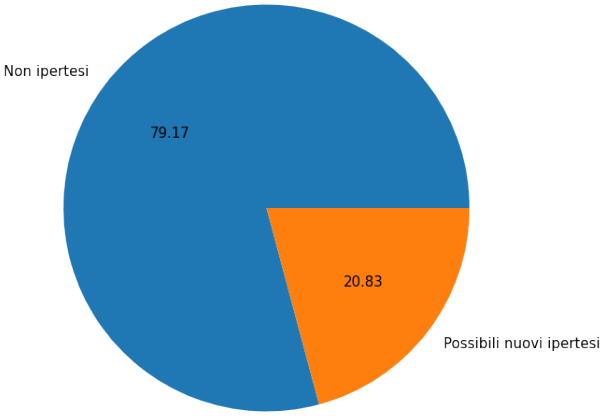

Figura 3.11: Percentuali sui possibili nuovi ipertesi e non ipertesi nell'insieme dei dati

di eventi cardiaci ischemici, si è fatto ricorso ad una codifica *one-hot* sulla base dei valori contenuti nelle stringhe delle risposte. Questo perché non solo diversi possono essere i parenti che in una stessa famiglia soffrono di ischemia cardiaca, ma anche perché nelle risposte in alcuni casi sono stati indicati più parenti contemporaneamente. È stato costruito un *dictionary* dove ad ogni nome di colonna che si voleva costruire *ex-novo* o mantenere dopo la trasformazione è stata associata una *list* di nomi di colonne i cui valori contribuivano al valore della colonna il cui nome è stato indicato come *key*. Questo metodo ha permesso di gestire sia i casi dei sinonimi dei nomi dei parenti - ad esempio "madre" e "mamma" -, sia i casi dei parenti indicati tramite nomi collettivi - ad esempio "genitori" al posto di "madre" e "padre" -, sia altri casi più particolari, come gli errori di battitura o i casi in cui l'intera famiglia fosse stata indicata come sofferente di ischemia cardiaca. Una volta utilizzate le associazioni così come definite, sono state eliminate tutte le colonne superflue, ovvero quelle contenute nelle *list* che fungevano da *values* nel *dictionary* stesso. Come ultimo passaggio, si è trattato di unire il risultato del nuovo *dataset* contenente le informazioni sui parenti con la settima risposta della prima domanda, che appunto chiedeva se nella propria famiglia esistessero persone che soffrissero di ischemia cardiaca. Sono stati usati i valori di quella risposta qualora un dato paziente non avesse indicato nessun parente ammalato, perciò quando tutte le colonne create con il metodo precedente presentassero valore pari a `False`. Nel fare questo, è stata considerata anche la colonna `ALTRO`, che catturava tutti quei casi in cui è stato indicato un parente che però non è stato possibile individuare con chiarezza. Dopodiché, è stata eliminata la colonna

della risposta perché integrata nel nuovo *dataset* che contiene le informazioni sui parenti, che è stato infine unito alle altre variabili.

3.4 Preprocessing dei dati

Si è deciso di usare la zona d'Italia di provenienza dei questionari come approssimazione per la zona di provenienza dell'intervistato stesso, anche se potrebbe non essere sempre una corretta assunzione. Alcune persone potrebbero trovarsi in determinate zone del Paese per necessità di lavoro o perché in vacanza, ma non necessariamente provengono da quella specifica zona. Questo ha significato dover eliminare tutti i questionari raccolti fuori dall'Italia, dato che non era possibile effettuare una buona assunzione sulla provenienza dell'intervistato. Mantenere questi questionari avrebbe voluto dire anche lasciare spazio alla possibilità che gli intervistati potessero essere non italiani, il che avrebbe significato considerare nel computo anche persone provenienti da gruppi etnici differenti che avrebbero potuto falsare l'analisi, seppure in piccola quantità.

Dall'insieme dei dati finale sono state eliminate tutte le istanze in cui il sesso era non dato, visto che non è possibile trovare un valore migliore di un altro che possa essere usato per imputare i valori mancanti.

Per quanto riguarda l'età degli intervistati, non essendo presenti *outlier*, non è stata eliminata alcuna istanza basandosi su di essi. Questa variabile è stata comunque utilizzata per andare ad individuare le istanze del problema di interesse, infatti ha permesso di eliminare tutti gli intervistati con età inferiore ai 18 anni. In questo modo è stato possibile condurre l'analisi dei dati sui soli maggiorenni, come ci si era prefissato. Per quanto riguarda le variabili di altezza, peso e battito cardiaco, prima di effettuare la loro analisi sono stati imposti dei limiti arbitrari, ma ragionevoli, sui valori della variabile, come già parzialmente indicato in precedenza. È stato fatto perché in queste erano presenti degli *outlier* il cui ordine di grandezza era molto maggiore di quello della stragrande maggioranza delle altre istanze, chiaro errore di lettura da parte del lettore ottico che ha digitalizzato i questionari. Questo portava ad una peggiore visualizzazione dei dati nei grafici nonché i valori della deviazione standard ad essere molto più grandi del dovuto e a costruire delle curve gaussiane di approssimazione peggiori. In ogni caso, tutti i valori eliminati in questo modo sarebbero stati comunque rimossi dalla successiva pulizia dei dati. I valori di *cut-off* arbitrari scelti in questo modo sono stati 120 cm e 220 cm per l'altezza, 30 kg e 160 kg per il peso, mentre per le misurazioni di frequenza cardiaca sono stati 20 bpm e 250 bpm. Per tutte le variabili numeriche originali e il B.M.I. l'eliminazione degli *outlier* è stata effettuata, al contrario di quanto fatto per

la variabile dell'età. La tecnica utilizzata è stata sempre la stessa e ha coinvolto il fatto che tutte potevano essere approssimate sufficientemente bene dalla curva gaussiana così come descritta in precedenza. I valori di *cut-off* scelti sono stati quelli a distanza $\pm 3\sigma$ dal centro della gaussiana, ovvero dal valore di μ della stessa. In particolar modo, per le variabili che rappresentavano le misurazioni di pressione sistolica, di pressione diastolica e di frequenza cardiaca, una volta ottenuti i *cut-off* per le tre misurazioni, i valori limite finali sono stati calcolati come la media di questi e sono stati applicati identicamente sulle tre misurazioni. In questo modo non si sono resi del tutto inutili i dati della prima misurazione di tutti e tre i parametri. Operando in questo modo, siamo stati sicuri di mantenere circa il 99,7% dei valori nel *dataset* per ciascuna variabile. Questo ha permesso di eliminare molti *outlier*, salvandone però altri più vicini ai valori ritenuti validi per la distribuzione. Non è stata effettuata alcuna eliminazione degli *outlier* a partire dalla variabile della frequenza cardiaca media. Per tutte le variabili numeriche meno le misurazioni di frequenza cardiaca e la frequenza cardiaca media, ovverosia età, altezza, peso, *Body Mass Index* e misurazioni di pressione sistolica e diastolica, i valori non dati sono stati semplicemente eliminati. Questo perché non c'è stato bisogno di recuperare istanze scartate: ne sono rimaste disponibili a sufficienza tali da rendere qualsiasi aggiunta di fonti di errore nei dati in cambio di un maggior numero di istanze superflua. Questo è vero a maggior ragione per le variabili di pressione sistolica e diastolica, dove imputare i valori utilizzando la media o la mediana o simili avrebbe significato un aumento in percentuale di persone che non sono possibili ipertesi, rendendo il problema di classificazione ancor più sbilanciato di quanto non fosse e rendendo perciò i modelli meno accurati. Chi è iperteso infatti, com'è facile immaginare, ha valori di pressione sistolica e diastolica superiori alla media. Diverso invece il discorso per quanto riguarda la frequenza cardiaca. Infatti, essendo state sostituite le tre misurazioni con la media delle ultime due, è sembrato sensato sostituire ai valori mancanti, in ciascuna delle due variabili, la media dei valori della misurazione stessa. In questo modo la variabile della frequenza cardiaca media, lì dove le ultime due misurazioni della frequenza cardiaca presentavano uno o più valori assenti, assumeva un valore molto vicino alla sua stessa media, essendo ottenuta tramite una media di valori che non si discostano molto dalla loro rispettiva media. In particolare, nel caso in cui sia la seconda che la terza misurazione presentino un valore assente, il valore assunto dalla frequenza cardiaca media sarà esattamente la sua stessa media. Infatti, denotando con $\overline{FC}$ la frequenza cardiaca media, con FC_2 e FC_3 rispettivamente la seconda e la terza misurazione di frequenza cardiaca, allora è vero che

$$\overline{FC} = \frac{FC_2 + FC_3}{2}$$

ma questo implica che

$$E[\overline{FC}] = E[\frac{FC_2 + FC_3}{2}] = \frac{E[FC_2 + FC_3]}{2} = \frac{E[FC_2] + E[FC_3]}{2}$$

per proprietà di linearità del valore atteso di una variabile aleatoria. Da qui è derivato quindi il fatto che imputare i valori mancanti nelle variabili originali con la media porta nella variabile risultato ad avere in corrispondenza di quei valori la media o un valore vicino ad essa.

Le variabili inerenti alle domande 1, 5 e 6, ovverosia quelle con risposte non mutualmente esclusive, data la loro codifica discussa in precedenza, sono state facilmente convertite in variabili binarie dove a 0 è stato associato il valore `False` e ad 1 è stato associato `True`. Per quanto riguarda invece le altre domande, che prevedevano una sola risposta tra le possibili, sarebbe stato possibile convertire il tipo delle variabili collegate a queste domande in uno nominale. I suoi valori sarebbero stati le possibili risposte alla domanda stessa, se non fosse stato per i valori inammissibili presenti. Si ricorda infatti che in tutte queste domande era presente il valore 0, mentre nella domanda 3 erano presenti valori 3 e 5. Eliminare tutti questi possibili errori di lettura sarebbe stato troppo distruttivo nei confronti dei dati, poiché alla fine del processo ne sarebbero rimasti soltanto qualche migliaia, perciò si è optato per una strategia diversa. Si è aggiunto il valore "non so" a tutte le domande che non l'avevano già, in modo tale da far confluire in questo valore tutti quelli che non rientravano tra i previsti. È sembrata una buona strategia non solo perché ha permesso di non eliminare istanze, ma perché alcune domande già presentavano questo valore ed era perciò ammissibile per esse. In questo valore sono anche stati fatti ricadere i valori 3 e 5 della domanda 3, che non erano ammissibili sotto nessuna condizione, ma che fortunatamente comparivano soltanto in 51 istanze complessivamente. Tutti i valori "non so" di tutte le domande, veri o inseriti da noi in questo modo, sono stati poi eliminati durante la selezione delle *feature* rilevanti, come si vedrà in seguito. Com'è facile intuire, per quanto riguarda le variabili inerenti le risposte alle domande, quindi compresa la risposta sul parente che soffre di ipertensione, i valori mancanti non sono stati eliminati perché sono stati tutti imputati con un valore adeguato che è stato, come già detto, quello equivalente a "non so" oppure "altro".

Riassumendo le operazioni di *preprocessing* dei dati fatte, descrivendole in maniera più puntuale da un punto di vista numerico, occorre dire innanzitutto che il *dataset* originale conteneva 37.110 istanze. Successivamente sono state eliminate tutte quelle dei pazienti che hanno risposto affermativamente alla domanda in cui si chiedeva se fossero malati di ipertensione oppure no, che ha fatto scendere le istanze a 24.772. Sono stati poi eliminati i pazienti che provenivano da fuori Italia, che ha portato le istanze a 24.627, e che non

avevano nessun valore indicato per il sesso, che le ha portate a 23.116. L'eliminazione delle istanze dovuta al fatto che le persone rappresentate da esse avevano un'età inferiore a 18 anni ci ha portato a rimanere con 22.795 istanze. Le eliminazioni fatte per rimuovere gli *outlier* ci hanno portato a 22.622 istanze quando sono state applicate sulla variabile inerente l'altezza. Da quel valore siamo scesi a 22.404 istanze quando sono state applicate sulla variabile del peso e infine a 22.215, quando poi si è considerato il *Body Mass Index*. La rimozione dei valori NA dalle variabili delle misurazioni di pressione sistolica e diastolica ci ha lasciato 21.075 istanze, mentre quella per rimuovere le istanze dove la pressione sistolica era inferiore a quella diastolica ce ne ha lasciate 21.010. L'eliminazione degli *outlier* nelle variabili di pressione sistolica ci ha fatto mantenere nel *dataset* 20.584 istanze, poi quella nelle variabili di pressione diastolica 20.367 e infine quella nelle variabili di frequenza cardiaca 20.058. Con queste ventimila istanze rimaste si è poi proseguito con la vera e propria fase di costruzione dei modelli.

3.5 Analisi delle multicollinearità e selezione delle feature più rilevanti

Per osservare l'eventuale presenza di multicollinearità tra le variabili, si è utilizzata la *heatmap* di Seaborn visibile nella figura 3.12. I valori da essa considerati sono stati ottenuti tramite l'indice di correlazione di Pearson, che permette di calcolare quanto bene una correlazione lineare approssima la relazione che esiste tra una coppia di variabili. Sono stati presi in considerazione solo i valori superiori a 0,3 in valore assoluto, dato che valori più piccoli indicano una scarsa correlazione tra le variabili e valori più grandi visualizzerebbero nel grafico troppe poche variabili. Inoltre, non sono stati visualizzati i valori lungo la diagonale, essendo sempre pari ad 1 e perciò per nulla significativi. Nonostante siano stati applicati questi vincoli, i valori dell'indice di correlazione tra le variabili rimangono sempre molto bassi, segno che, almeno considerando le variabili a coppie, non esistono correlazioni lineari significative tra esse e perciò è molto poco probabile che siano presenti fenomeni di multicollinearità. Come si può notare, le correlazioni più significative si hanno tra le variabili che rappresentano i parenti che soffrono di ischemia cardiaca. Questo è dovuto a come le variabili sono state costruite, dato che in molti casi colonne diverse hanno ricevuto contributi nella loro costruzione a partire dalle stesse variabili. È ad esempio il caso delle variabili che rappresentano uno o due nonni malati di ischemia e una o due nonne malate di ischemia. Entrambe queste variabili infatti sono state costruite dalla variabile originale "`NONNI`", che si è ragionevolmente assunto indicasse quei casi dove una o entrambe le coppie di nonni

del paziente soffrissero di ipertensione. Questa variabile, avendo molti valori `True`, ha finito per incidere molto sulle distribuzioni di entrambe le variabili ottenute da essa.

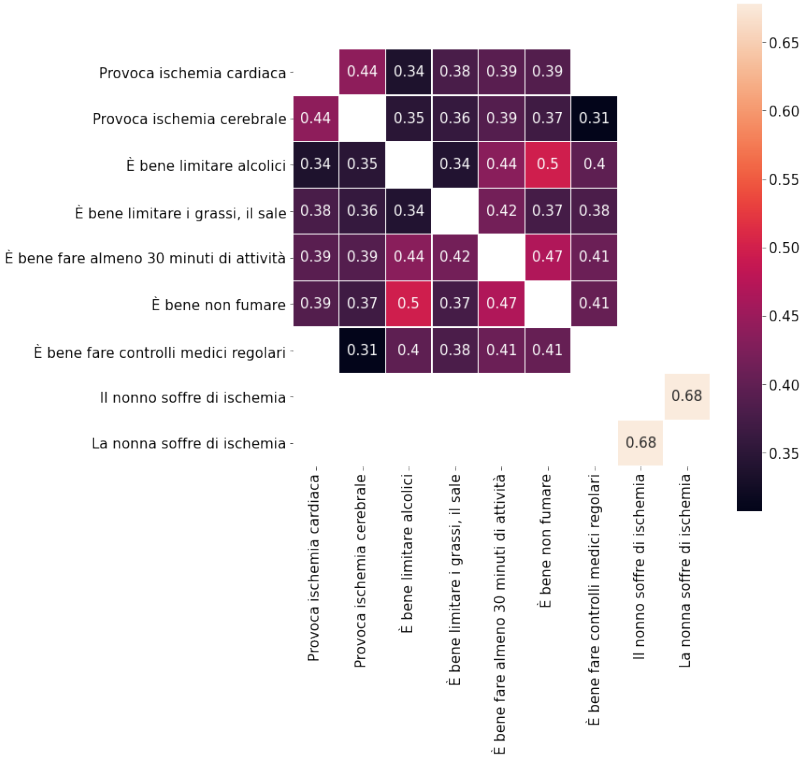

Figura 3.12: *Heatmap* dei valori dell'indice di correlazione di Pearson tra le coppie di variabili presenti nel *dataset*

Altre coppie di variabili che presentano correlazioni significative sono quelle costituite dalle risposte alle domande 5 e 6, ovverosia quelle di cultura generale sull'ipertensione che chiedevano quali erano le co-morbilità della stessa e quali sono i migliori mezzi per contrastarla. Evidentemente il questionario presentava molte risposte che si prestavano ad essere scelte insieme, come ad esempio "è bene non fumare" ed "è bene limitare gli alcolici" oppure "è bene non fumare" ed "è bene fare almeno 30 minuti di attività fisica al giorno". Molto spesso queste affermazioni vengono suggerite assieme in quelli che sono *vademecum* generalmente corretti per prevenire malattie di qualsiasi tipo. Delle altre variabili non c'è traccia, segno che non esistono altre correlazioni sufficientemente significative oltre a quelle facilmente intuibili.

A questo punto, è stata effettuata l'eliminazione delle variabili meno significative, e quindi la selezione di quelle più significative, attraverso l'uso della regolarizzazione L1. Si è innanzitutto suddiviso l'insieme dei dati in *training*

set e *test set* con una proporzione 70-30 e si sono poi addestrate una serie di regressioni logistiche "di prova" attraverso l'uso di una *grid search*. Per ciascun valore del parametro di regolarizzazione λ si sono ottenuti i valori delle metriche di valutazione attraverso l'uso della *cross-validation*, in modo da mitigare eventuali oscillazioni nelle metriche stesse dovute a variazioni nei dati. Dato che il problema era sbilanciato, si è deciso di applicare un coefficiente agli errori delle due classi pari alla metà del reciproco della percentuale di istanze appartenenti alla classe stessa. Questo è all'incirca equivalente ad effettuare *oversampling* della classe di minoranza fino ad avere entrambe le classi con lo stesso numero di istanze. Inoltre, non si è scelto come migliore il modello che aveva accuratezza più alta, ma bensì quello che aveva sensibilità più alta. In questo modo si è evitato che venisse scelto come modello migliore sistematicamente quello che classificava la maggior parte delle istanze come "non ipertesi", cioè la classe con il maggior numero di istanze. Questo è un problema comune dei *dataset* medici, dato che, sperabilmente, solo una piccola frazione della popolazione soffre di una data malattia. Trovato quindi il modello migliore secondo la metrica adottata, abbiamo osservato quali variabili ha eliminato e quali ha invece mantenuto. Le variabili eliminate sono state:

- È fumatore
- Soffre di insufficienza renale
- È diabetico
- Ha sofferto in passato di eventi ischemici cardiaci
- Soffre di colesterolo alto
- Ha sofferto in passato di eventi ischemici cerebrali
- L'ipertensione provoca l'ischemia cerebrale
- L'ipertensione provoca l'insufficienza renale
- L'ipertensione provoca l'insufficienza epatica
- L'ipertensione provoca cecità
- L'ipertensione provoca il diabete mellito
- È bene seguire una dieta con poche calorie e molte proteine
- È bene limitare il consumo di alcolici
- È bene bere un bicchiere di vino rosso al giorno

- È bene fare almeno 30 minuti di attività fisica al giorno
- È bene fare solo attività fisica intensa
- È bene non bere caffè
- È bene fare un controllo medico non appena compaiono i sintomi, ma non prima
- I bisnonni soffrono di ischemia cardiaca
- Un cugino soffre di ischemia cardiaca
- Una figlia soffre di ischemia cardiaca
- Un figlio soffre di ischemia cardiaca
- Un fratello soffre di ischemia cardiaca
- La mamma soffre di ischemia cardiaca
- Il marito soffre di ischemia cardiaca
- La moglie soffre di ischemia cardiaca
- Un nipote soffre di ischemia cardiaca
- Un nonno soffre di ischemia cardiaca
- Una zia soffre di ischemia cardiaca
- Una nonna soffre di ischemia cardiaca
- Una cugina soffre di ischemia cardiaca
- Un parente non tra i precedenti soffre di ischemia cardiaca
- Non sa se è stato ricoverato a causa dell'ipertensione
- Si misura la pressione meno di una volta l'anno
- Non sa quanto spesso si misura la pressione
- Non soffre di sonnolenza diurna
- Non sa se soffre di sonnolenza diurna
- Non sa se russa di notte

- Russa di notte
- Non soffre di apnee nel sonno
- Non sa se soffre di apnee nel sonno
- Soffre di apnee nel sonno
- Proviene dal nord Italia
- Proviene dal sud Italia

Mentre le variabili che sono state mantenute sono:

- Età
- È di sesso femminile
- L'ipertensione provoca ischemia cardiaca
- È bene seguire una dieta povera di grassi e sale, ma ricca di vitamine e fibre
- È bene non fumare
- È bene effettuare controlli regolari, anche se non si hanno disturbi
- Indice di massa corporea
- Il padre soffre di ischemia cardiaca
- Una sorella soffre di ischemia cardiaca
- Uno zio soffre di ischemia cardiaca
- Frequenza cardiaca media
- Non è mai stato ricoverato a causa dell'ipertensione
- È stato ricoverato a causa dell'ipertensione
- Si misura la pressione giornalmente
- Si misura la pressione mensilmente
- Si misura la pressione annualmente
- Soffre di sonnolenza diurna

- Non russa di notte
- Proviene dal centro Italia

Come si può notare, sono state eliminate molte delle risposte alle domande 1, 5 e 6. Se per la domanda 1 questo potrebbe essere interessante, non è invece una sorpresa per le domande 5 e 6, dato che si riferiscono a possibili, ma non necessari, cambiamenti nello stile di vita adottati dal paziente. Sono stati eliminati anche la stragrande maggioranza dei parenti che soffrono di ischemia cardiaca, probabilmente perché erano troppo pochi per indicare una qualche correlazione genetica dell'ipertensione. Notiamo anche che sono state eliminate tutte le variabili relative alle risposte "non so". Tra le variabili che sono state mantenute vediamo i classici dati antropometrici come età, sesso, B.M.I., oltre alla frequenza cardiaca media che ragionevolmente possiamo pensare associata a fenomeni ipertensivi. Inoltre, come potevamo aspettarci, essersi fatti ricoverare a causa dell'ipertensione e la frequenza di misurazione della pressione sono correlate alla presenza di ipertensione nel soggetto intervistato. Sono state quindi eliminate da *training* e *test set* tutte le variabili che sono state reputate in questo modo non sufficientemente significative.

3.6 Costruzione e valutazione dei modelli di apprendimento

In fase preliminare alla costruzione dei modelli di apprendimento si è voluta quantificare la complessità del problema tramite algoritmi di riduzione della dimensionalità considerando tutte le variabili del problema, anche quelle scartate nella fase precedente. Per fare questo, è stato utilizzato l'algoritmo "t-Stochastic Neighbor Embedding" per "comprimere" i dati in due dimensioni e poterli visualizzare, il cui risultato si può vedere nella figura 3.13. Non è stato possibile definire un *cluster* ben definito di istanze negative o positive, segno che non esiste un modo "semplice" per suddividere i dati nelle due classi. Questo significa anche che il problema è complesso e non necessariamente sarà possibile trovare una buona soluzione.

Tutti i modelli sono stati poi costruiti in maniera analoga a come è stata costruita la regressione logistica "di prova" per l'estrazione delle *feature* rilevanti. Questo significa che ad ogni modello è stato applicato il bilanciamento delle classi con i valori scelti in precedenza, sono state usate in congiunzione *grid search* e *cross-validation* e la metrica che è stata adottata per individuare il modello migliore, ovverosia quella che si è cercata di massimizzare, è stata la sensibilità del modello stesso. Unica eccezione è stata "XGBoost", per il quale

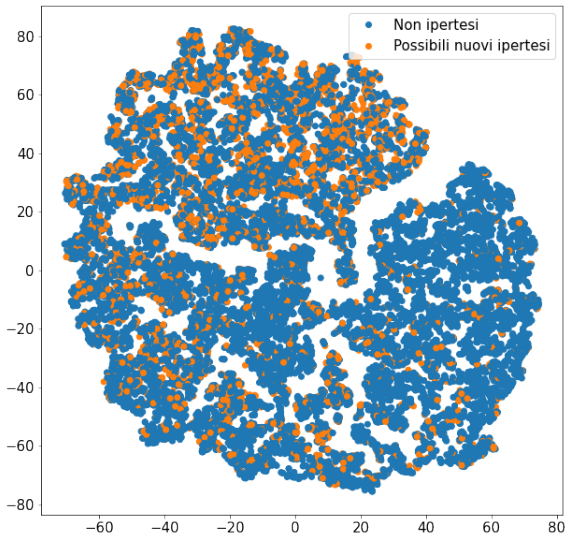

Figura 3.13: Visualizzazione del risultato dell'algoritmo t-SNE sull'insieme dei dati considerando tutte le variabili

non era possibile specificare un peso per ciascuna delle classi del problema, ma occorreva indicare il rapporto tra i pesi delle classi positiva e negativa. Si è scelto come valore di questo parametro quello suggerito dalla documentazione del metodo stesso, ovvero il rapporto tra il numero delle istanze negative e il numero delle istanze positive. Una volta trovati gli iperparametri migliori, per ottenere i veri valori di accuratezza, specificità e sensibilità in corrispondenza di essi, si è addestrato un modello con quegli stessi iperparametri ma sull'intero *training set*, in modo tale da fargli sfruttare tutta la conoscenza di cui disponeva per effettuare le proprie predizioni.

Il primo metodo utilizzato è stato "Decision Tree". Data la sua semplicità, e quindi la sua rapidità di esecuzione, con esso è stato possibile utilizzare un filtro per la costruzione di *feature* polinomiali dalle variabili originali. In questo modo la complessità del modello prodotto è aumentata e con essa anche il suo *score*. Non a caso, il modello migliore secondo le metriche indicate precedentemente è stato quello che utilizzava un filtro per *feature* polinomiali di terzo grado, il grado più alto utilizzato all'interno della griglia di ricerca. Inoltre, il modello migliore si è ottenuto in corrispondenza del valore massimo indicato per il parametro che determina quanti elementi devono essere presenti in un nodo per poter effettuare una divisione dello stesso, ovvero 8. Tutti i modelli migliori avevano il valore massimo specificato come parametro che determina quanti elementi almeno devono essere presenti in ciascuna foglia dell'albero, cioè 7. Inoltre, tutti i modelli migliori usavano come numero di variabili da considerare per la migliore divisione di un nodo il logaritmo in

base due del numero delle *feature*. Il modello ottenuto ha una sensibilità del 61,3%, una specificità del 54,9% e un intervallo di accuratezza compreso tra il 55,9% e il 58,4% con confidenza pari al 95%. Qui di seguito si mostra la sua matrice di confusione.

	Negativo	Positivo
Negativo	2615	2149
Positivo	430	824

Il passo successivo è stato quello di utilizzare "Random Forest". Il miglior modello ottenuto con questo metodo ha come iperparametri per i singoli alberi gli stessi di quello costruito con "Decision Tree", mentre come numero di alberi nella foresta ha il valore più basso specificato nella *grid*, ovverosia 100. Questo modello ha una sensibilità del 57,5%, una specificità del 75,7% e un intervallo di accuratezza compreso tra il 70,8% e il 73,1% con una confidenza del 95%. Qui di seguito si mostra la sua matrice di confusione.

	Negativo	Positivo
Negativo	3608	1156
Positivo	533	721

A seguire, si è utilizzato uno dei metodi più noti nella classificazione che è la regressione logistica. In questo caso, il modello migliore si è rivelato quello che utilizzava come regolarizzazione "Elastic Net" con un parametro della stessa relativamente alto, ovvero pari a 100. Il modello migliore aveva inoltre come parametro che esprime il rapporto tra regolarizzazione L1 ed L2 il valore 0,5, segno che ha svolto un ruolo importante sia il contenimento del valore assoluto dei coefficienti, sia l'eliminazione delle variabili superflue. Da ultimo, il modello migliore aveva come valore per la tolleranza nel criterio di terminazione la più piccola indicata, ovvero 10^{-8}. Questo modello ha mostrato una sensibilità del 69,7%, una specificità del 65,1% e un intervallo di accuratezza tra il 64,9% e il 67,3% con una confidenza del 95%. Qui di seguito si mostra la sua matrice di confusione.

	Negativo	Positivo
Negativo	3102	1662
Positivo	380	874

Dopo la regressione logistica, è stato naturale pensare di utilizzare un metodo che effettua una classificazione non lineare, che quindi è capace di costruire dei modelli più complessi attraverso *feature* non polinomiali, in particolar modo attraverso l'uso di funzioni *kernel*: le *Support Vector Machines*. Il modello migliore ottenuto tramite questo metodo utilizzava una funzione *kernel* di tipo "Radial Basis" e un parametro di regolarizzazione L2 relativamente alto,

ovverosia 100, uguale a quello del miglior modello di regressione logistica. La sensibilità di questo modello è stata pari al 77,8% mentre la sua specificità è stata del 56,2%. L'intervallo di accuratezza era compreso tra il 59,5% e il 61,9% con una confidenza pari al 95%. Qui di seguito si mostra la sua matrice di confusione.

	Negativo	Positivo
Negativo	2299	2465
Positivo	199	1055

Si è voluto provare anche un metodo al di fuori delle librerie *scikit-learn*, nella fattispecie "XGBoost". Il modello migliore ottenuto aveva come parametro di *learning rate* il valore 0,5 e come parametri delle regolarizzazioni L2 ed L1 il valore 100, valore discretamente alto ma soprattutto non dissimile da quello individuato nella regressione logistica e nell'uso delle "Support Vector Machines". Per di più, sempre come nella regressione logistica, il miglior modello si è avuto in corrispondenza di un egual peso assegnato alla regolarizzazione L1 ed L2. La sua sensibilità è stata del 72,6%, la specificità del 61,8% mentre l'intervallo di accuratezza era compreso tra il 62,8% e il 65,2% con una confidenza pari al 95%. Qui di seguito si mostra la sua matrice di confusione.

	Negativo	Positivo
Negativo	2942	1822
Positivo	344	910

Ultimo, ma non per questo meno importante, è stato l'utilizzo del metodo "Linear Discriminant Analysis", che anch'esso è stato combinato con il filtro per la creazione di *feature* polinomiali. In particolar modo, quest'ultimo è stato utilizzato per effettuare una classificazione polinomiale con polinomi di secondo e terzo grado. A parità degli altri parametri presenti nella griglia, l'utilizzo di un modello più complesso, nello specifico l'uso di un polinomio di grado tre, ha sempre premiato, dato che molti dei migliori modelli presentano questo tipo di trasformazione delle variabili. Inoltre, tutti i migliori modelli si basano sull'utilizzo degli autovalori e in particolar modo i primi due modelli utilizzano come valore per il parametro di riduzione quello più alto possibile, ovvero 0,9. Il miglior modello prodotto aveva una sensibilità del 61,5%, una specificità del 72,4% e l'intervallo di accuratezza era compreso tra il 69,0% e il 71,3% con una confidenza del 95%. Qui di seguito si mostra la sua matrice di confusione.

	Negativo	Positivo
Negativo	3449	1315
Positivo	483	771

Infine, si mostra la tabella riassuntiva che mette a confronto le sensibilità, le specificità e gli intervalli di accuratezza dei migliori modelli ottenuti dai metodi elencati. Tra i modelli è presente anche quello definito "casuale", ovverosia che effettua le proprie predizioni senza essere stato costruito sul *dataset*, ma in un modo definito da noi a priori. Come si può intuire dalla tabella, è stato costruito per assegnare ad ogni nuova istanza la classe più frequente nell'insieme dei dati, ovvero quella dei negativi, dei "non ipertesi". Appare quindi chiaro perché la metrica da massimizzare non poteva essere l'accuratezza o la specificità, dato che per esse il modello casuale è il modello migliore, ma tuttavia per noi è quello con i risultati più scarsi, con una sensibilità dello 0%.

Modello	Accuratezza		Sensibilità	Specificità
	Inferiore	Superiore		
Casuale	0.781179	0.801698	0.000000	1.000000
Random Forest	0.707852	0.730551	0.574960	0.757347
LDA	0.689539	0.712662	0.614833	0.723971
Decision Tree	0.558907	0.583905	0.657097	0.548908
Regressione Logistica	0.648623	0.672541	0.696970	0.651134
XGBoost	0.627867	0.652113	0.725678	0.617548
SVM	0.594607	0.619280	0.777512	0.562133

3.7 Interpretazione della conoscenza appresa

Come si è potuto osservare, il modello che presenta una sensibilità più alta è quello prodotto dal metodo delle "Support Vector Machines", ma è anche quello che ha anche una specificità che è la seconda tra le più basse. Inoltre, i coefficienti estratti da questo modello sono di difficile interpretazione data la sua complessità intrinseca. Mentre invece i modelli che bilanciano meglio il rapporto tra sensibilità e specificità, pur mantenendo la prima sufficientemente alta, sono i secondi migliori due, ovvero quelli prodotti dalla regressione logistica e da "XGBoost". Analizzeremo perciò le informazioni che questi due modelli sono riusciti ad estrarre.

In prima battuta possiamo analizzare l'importanza che i due modelli hanno assegnato a ciascuna delle *feature* con cui hanno lavorato. L'importanza che il modello ottenuto con "XGBoost" assegna ad ogni *feature* è visibile nella figura 3.14 ed è il valore del cosiddetto "gain". Questa metrica permette di mostrare il miglioramento relativo a cui porta una divisione su un nodo operata da una specifica variabile sullo "F-score", miglioramento che è quindi calcolato per tutte le divisioni di tutti gli alberi contenuti nel modello. Lo "F-score" è invece una possibile metrica per il calcolo dell'accuratezza del modello. Dalla

definizione si deduce perciò che maggiore è il "gain", maggiore è il contributo che una variabile dà nel migliorare l'accuratezza del modello.

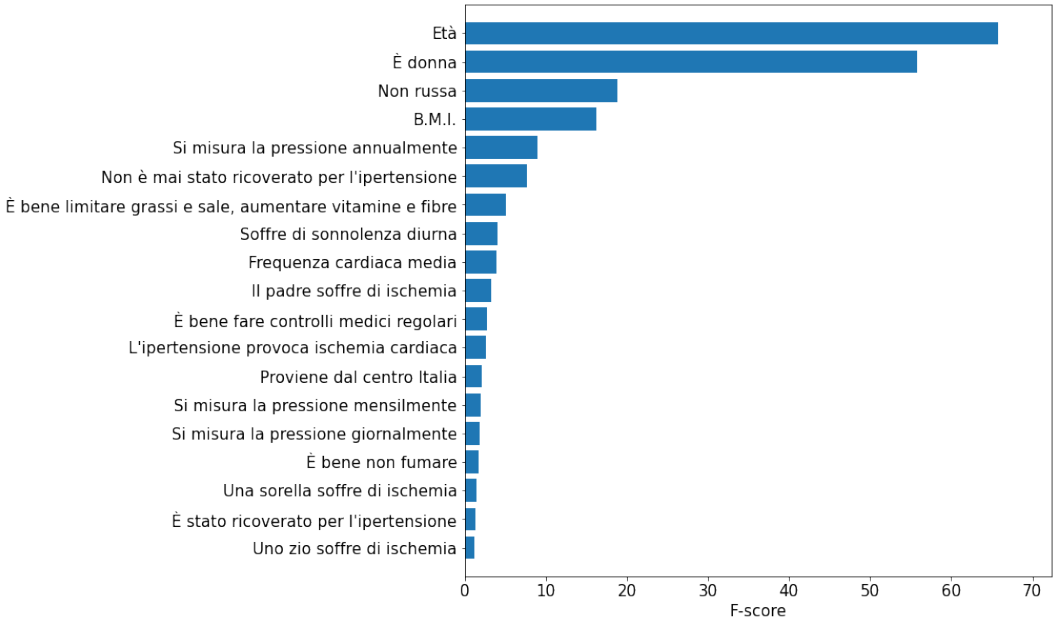

Figura 3.14: Importanza espressa come "*gain*" data alle singole variabili dal miglior modello ottenuto con "XGBoost"

Nella figura 3.15 possiamo invece vedere l'importanza che la regressione logistica dà alle variabili. In questo caso i valori che sono mostrati nel grafico rappresentano i coefficienti dell'iperpiano di separazione individuato come soluzione del problema dalla regressione logistica. Quanto più i valori sono grandi, tanto più quella variabile avrà un peso maggiore nel determinare il risultato. Se il coefficiente è positivo allora la variabile contribuisce a spostare il risultato verso un valore positivo, ovvero porta il modello a classificare le istanze come parte della classe dei positivi, e viceversa. Come si può notare, le prime tre variabili più importanti sono tra le più importanti anche nel grafico di "XGBoost" e sono presenti all'interno di quest'ultimo nello stesso ordine. Evidentemente i modelli, pur con le loro differenze, sono d'accordo: l'età di una persona è un fattore che incide positivamente nello sviluppo dell'ipertensione, così come l'indice di massa corporea. Al contrario, le donne sono meno propense a sviluppare l'ipertensione rispetto agli uomini. Queste informazioni ci dicono che questi modelli sono sulla buona strada: i precedenti sono fatti da tempo acquisiti all'interno della comunità medica. L'importanza del sesso, dell'età e dell'indice di massa corporea è testimoniato anche da un articolo apparso su Plos One nel luglio 2013 [7]. Esso mette a confronto 15 modelli di

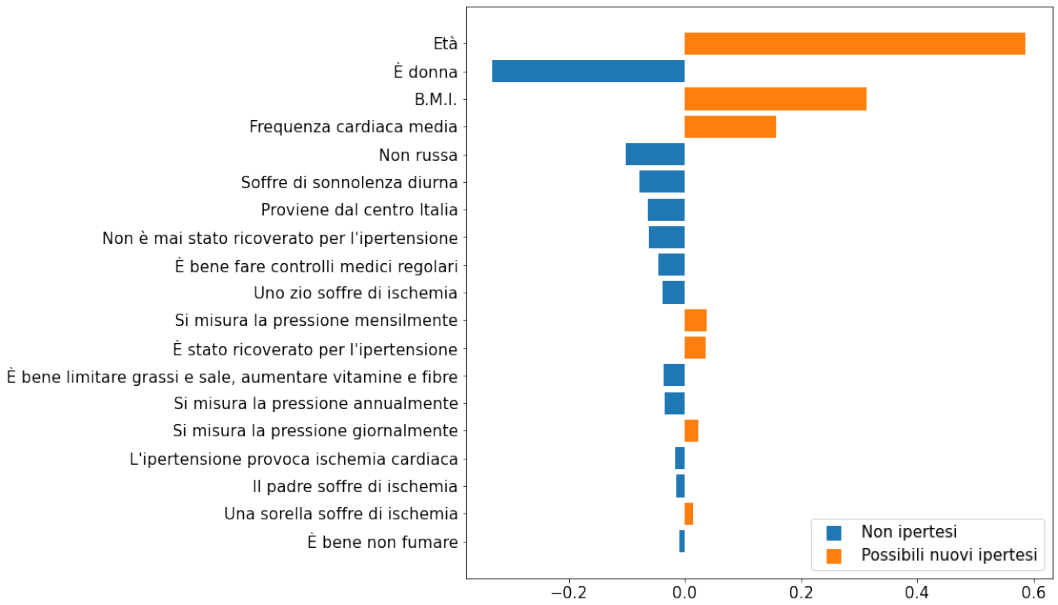

Figura 3.15: Importanza espressa come valori dei coefficienti dell'iperpiano data alle singole variabili dal miglior modello ottenuto con regressione logistica

predizione estratti da 11 articoli differenti ottenuti attraverso PubMed ed EMBASE. Nella maggior parte dei modelli esaminati, queste tre variabili comparivano come predittori e in tutti e 15 almeno una delle tre appariva. Sempre tra le variabili molto importanti in entrambi i grafici vediamo anche "non russa" e "non è mai stato ricoverato per l'ipertensione". Effettivamente se si è stati ricoverati a causa dell'ipertensione con buona probabilità si soffre di ipertensione, così come chi non russa presenta un sintomo in meno tra quelli collegati all'ipertensione e con minor probabilità ne soffre. Un'altro dettaglio interessante è il fatto che se per la regressione logistica la frequenza cardiaca media è una variabile molto interessante, per "XGBoost" non lo è. Diversamente da quanto potevamo aspettarci, chi soffre di sonnolenza diurna meno probabilmente soffre anche di ipertensione, che è il contrario di quanto indicato dai manuali medici. Infine, anche in questo caso diversamente dalle aspettative, secondo il modello di regressione logistica più frequentemente ci si misura la pressione e più probabilmente si soffre di ipertensione. Potrebbe essere che chi è e si sente più a rischio, ma tuttavia non pensa di essere ammalato, tenda a tenere d'occhio con più scrupolosità le misurazioni di pressione. Un ultimo dettaglio degno di nota è che, benché sia da entrambi i modelli considerata una variabile relativamente poco interessante, compare "per prevenire l'ipertensione è bene non fumare". Il predittore inerente al fumo è infatti anch'esso tra le variabili maggiormente presenti nei modelli analizzati dal già citato articolo di Plos One. È bene comunque ricordare che la variabile che indicava se

un paziente fosse fumatore o meno, indipendentemente dal fatto che sapesse che fumare aumentasse il rischio di sviluppare l'ipertensione, è stata eliminata in precedenza nella fase di selezione delle *feature*. Nei modelli presentati nel summenzionato articolo viene molto spesso indicato anche come predittore la presenza o meno dell'ipertensione nei genitori del paziente. Nel nostro caso si è analizzata la storia parentale dell'ischemia, ma il fatto che una delle variabili finali, pur con la sua modesta rilevanza, sia se il padre soffra o meno di ischemia, crea un qualche sorta di collegamento tra le due cose.

Come ultimo grafico, si è osservato quello che permette di visualizzare i risultati del modello prodotto grazie a "Linear Discriminant Analysis", che è contenuto nella figura 3.16.

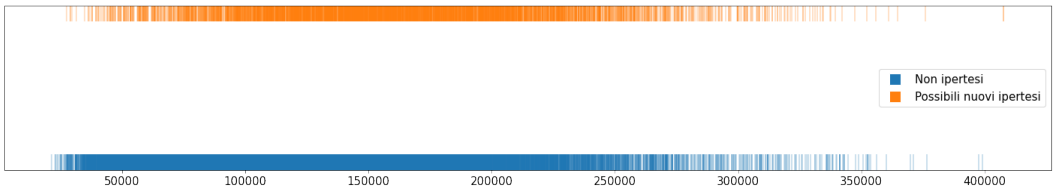

Figura 3.16: Le istanze del *dataset* così come il modello creato da "Linear Discriminant Analysis" le pone lungo la componente che ha costruito nel suo addestramento, divise per classe di appartenenza

Questo grafico mostra come vengono mappati tutti i valori presenti nel *dataset* lungo la componente, o asse, che il miglior modello ottenuto tramite "Linear Discriminant Analysis" è riuscito a costruire durante il suo addestramento. Come si può vedere, anche se vicino agli estremi esistono dei valori limite oltre i quali sono più presenti istanze appartenenti all'una o all'altra classe, esse sono nella stragrande maggioranza dei casi sovrapposte lungo l'asse. Quindi non è possibile andare ad individuare con chiarezza un "taglio" lineare lungo questa nuova dimensione che riesca a separare bene le istanze nelle due classi, così come ci aspettavamo e come i risultati ci hanno poi confermato.

Infine, si mostra il primo degli alberi decisionali della foresta costruita dal miglior modello ottenuto con "XGBoost". Questo per poter dare un'idea di come gli alberi sono costruiti e di come si traducono i ragionamenti fatti precedentemente su questo modello nella realtà. La figura che mostra questo albero è la 3.17. I nodi contengono le valutazioni da effettuare su una nuova istanza per poter raggiungere le foglie mentre si scende lungo l'albero, mentre in queste ultime è contenuta la probabilità percentuale che un'istanza che le raggiunge appartenga alla classe dei positivi. Nella realtà però le foglie contengono un numero che non è questa probabilità, ma è il cosiddetto "*logit score*". Per di più, la probabilità mostrata nel grafico è calcolata considerando il solo

valore presente nelle foglie di quest'albero, mentre invece la vera probabilità di appartenenza alla classe dei positivi si calcola elaborando la somma dei "*logit score*" nelle foglie raggiunte in tutti gli alberi della foresta. In questo caso, visto che dell'intera foresta si considera un albero solo, per semplicità e per capire meglio a colpo d'occhio come l'albero è strutturato, si è compiuta questa piccola incorrettezza. Il colore arancione di una foglia indica una maggiore probabilità di appartenenza ai "possibili ipertesi", il colore blu ai "non ipertesi", mentre il grigio indica un'indecisione tra le due classi, quindi una probabilità del 50% o un "*logit score*" pari a 0.

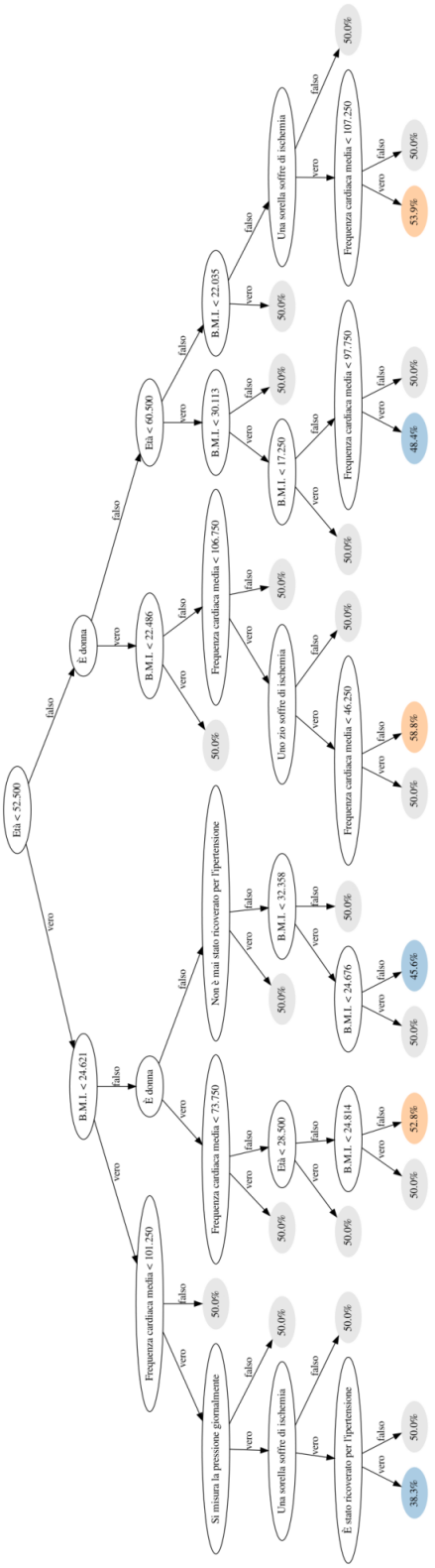

Figura 3.17: Il primo degli alberi decisionali della foresta che compone il miglior modello costruito tramite “XGBoost”

Capitolo 4

Sviluppo di un chatbot

In questo capitolo si discuterà del secondo dei due problemi evidenziati in precedenza, ovvero quello inerente allo sviluppo di un sistema per il supporto al paziente iperteso nella cura contro la sua malattia. L'idea è stata quella di costruire un *chatbot* che potesse interagire in maniera naturale con il paziente, affinché l'utente si sentisse maggiormente coinvolto e fosse così più propenso a collaborare con il sistema.

4.1 Analisi dei requisiti funzionali

Gli incontri che sono stati fatti con il dottor Pengo, come già detto, sono stati utili per delineare le specifiche del sistema che si è poi posto come soluzione al problema di supportare i pazienti ipertesi durante la loro cura. Egli ha fatto emergere la necessità di sviluppare un sistema che fornisca risposte adeguate ad un insieme limitato di richieste poste dall'utente in linguaggio naturale, altresì detto *chatbot*, per la piattaforma di messaggistica istantanea "Telegram". I primi due messaggi a cui deve saper rispondere sono "\start" e "\help", i quali sono imposti dalla API dei *bot* di Telegram perché il sistema sia conforme alle specifiche che la piattaforma di messaggistica stessa fornisce. Dopodiché, gli altri messaggi previsti sono quelli utili allo scopo per cui il *chatbot* è stato progettato. Deve innanzitutto essere capace di fornire le informazioni utili per potersi misurare correttamente la pressione. Conseguentemente, deve poter permettere all'utente di registrare le proprie misurazioni di pressione arteriosa, le quali contengono sempre il valore di pressione sistolica e il valore di pressione diastolica e possono opzionalmente contenere anche quello di frequenza cardiaca. Queste misurazioni possono essere comunicate al *chatbot* in formato testuale, e saranno quindi estratte dal messaggio di richiesta di registrazione stesso, oppure attraverso una fotografia dello schermo del misuratore di pressione. Se il valore registrato è al di sopra dei valori

ritenuti "target" per un paziente iperteso, ovverosia 135 mmHg per la pressione sistolica e 85 mmHg per la pressione diastolica, il *chatbot* dovrà inviare un messaggio di avvertimento il cui contenuto dipenderà dalla media mensile delle misurazioni. Se anche quest'ultima eccede i valori target, dovrà essere consigliato all'utente di parlarne con il proprio medico, mentre se invece la media mensile è in linea con i target lo si dovrà rassicurare. Una volta registrata una nuova misurazione, dovrà essere possibile recuperare tutte quelle precedentemente effettuate, oppure recuperare la media giornaliera, settimanale o mensile di queste ultime. Poiché mostrare tutte le misurazioni raccolte dall'utente è stato pensato principalmente per il medico che dovrà analizzarle, si vuole che in risposta a questa richiesta da parte dell'utente venga mostrato anche un grafico per ciascuno dei parametri raccolti. Questi tre grafici, oltre ai valori registrati nelle singole misurazioni, devono indicare anche il valore medio di ciascuno dei parametri.

Il sistema non dovrà essere solamente reattivo, ma anche proattivo, inviando adeguati promemoria. Nella fattispecie dovrà essere ricordato mensilmente all'utente di prendere la propria terapia e settimanalmente di misurarsi la pressione. Per quanto riguarda quest'ultimo caso, si vuole inviare il promemoria al paziente solo nel caso in cui siano passati troppi giorni dall'ultima misurazione effettuata. In particolare, dovrà essere inviato un messaggio dopo due giorni dalla data dell'ultima misurazione, dopodiché dopo tre giorni dalla precedente notifica e infine dopo quattro giorni dall'ultima notifica ricevuta. L'obiettivo è di ottenere almeno due misurazioni di pressione a settimana dall'utente senza che esso riceva un numero di promemoria troppo elevato. Dall'analisi dei requisiti indicati è nato il diagramma dei casi d'uso presente nella figura 4.1.

4.2 Analisi dei requisiti non funzionali

Il sistema dovrà essere *user-friendly*, dato che si vuole fare in modo che la platea degli utilizzatori sia la più ampia possibile. Occorre anche considerare che l'età e l'alfabetizzazione tecnologica di chi utilizzerà il sistema sarà molto varia, il che rende l'aspetto della *user experience* ancora più complesso e allo stesso tempo importante. Inoltre, a causa dell'importanza che il sistema riveste per la salute delle persone, si vuole che il suo tempo di sviluppo sia breve, affinché possa essere dato subito in mano ai potenziali utenti e possano quindi valutare la sua bontà. I messaggi che il sistema invierà non dovranno essere ambigui, ma anzi dovranno indicare chiaramente che cosa si vuole che l'utente faccia e come quest'ultimo dovrà portare avanti le successive interazioni con il sistema. Il *chatbot* deve essere capace di comprendere anche i messaggi formulati in maniera complessa e articolata, purché sia chiara la richiesta che l'utente

sta rivolgendo al sistema. Infine, la latenza di elaborazione dei messaggi deve essere sufficientemente breve da non risultare in un'attesa sconveniente per il paziente.

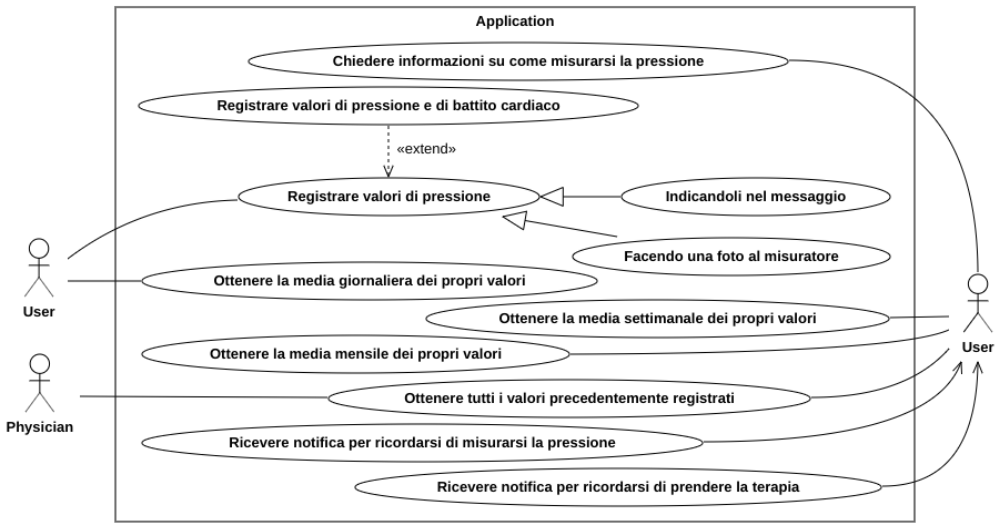

Figura 4.1: Diagramma dei casi d'uso costruito a partire dalle specifiche funzionali

4.3 Progettazione architetturale del sistema

L'architettura di questo sistema può essere più facilmente compresa osservando il diagramma delle classi che la descrive così come contenuto nella figura 4.2. L'architettura del sistema ha radice nel componente "Application", che rappresenta l'applicazione nel suo complesso. In pratica esso è il "Controller" del sistema, con lo stesso significato che questo termine riveste nel pattern architetturale "Model-View-Controller". Questo componente ha infatti la responsabilità di costruire ed avviare tutti gli altri e di interagire con essi nel momento in cui ciò si rende necessario, incapsulando la logica di esecuzione dell'intero *software*. Il cuore del modello su cui si basa l'applicazione è però il componente "MessageDispatcher". Come suggerisce il nome, è quello a cui vengono diretti tutti i messaggi provenienti dal *client* di Telegram per poi essere gestiti tramite l'invio di adeguata risposta. Questo componente deve servirsi di un *database* e di due interpreti per il contenuto dei messaggi, uno per quelli testuali e uno invece per quelli composti da un'immagine. Una volta ricevuto un messaggio, da questo il MessageDispatcher costruirà uno "UserMessage", che è la rappresentazione di un messaggio dell'utente. Questa entità avrà

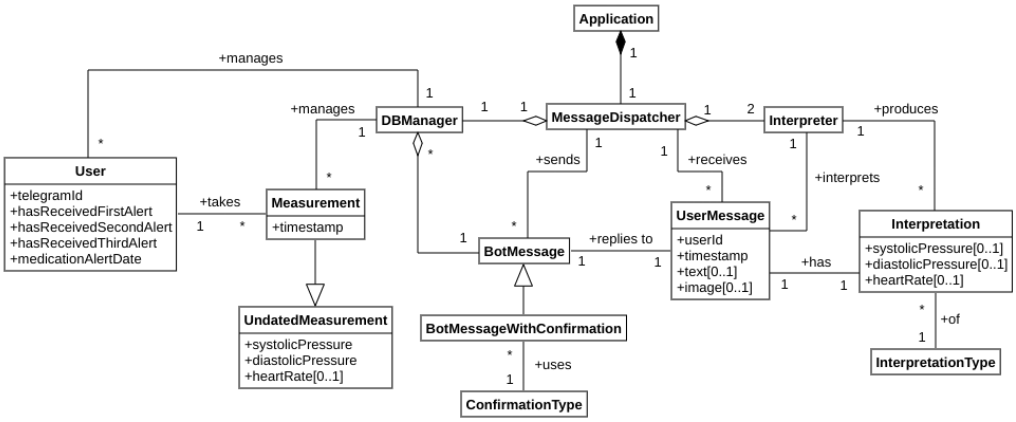

Figura 4.2: Diagramma delle classi che rappresenta il dominio applicativo del sistema

tutte le proprietà che caratterizzano il messaggio stesso, come l'identificativo dell'utente che lo ha inviato, il *timestamp* dell'istante in cui è stato inviato, il suo testo, se è un messaggio testuale, o l'immagine contenuta in esso, se invece è un messaggio multimediale. Creato lo UserMessage, il passo successivo è interpretarlo passandolo allo "Interpreter" adeguato a seconda del suo contenuto. L'Interpreter produce poi un'interpretazione, rappresentata dall'entità "Interpretation", la quale ha un tipo le cui istanze sono le possibili richieste che l'utente può rivolgere al *chatbot*. Quella che sta venendo rivolta, come precedentemente evidenziato dalle specifiche, potrebbe anche essere di salvare una misurazione di pressione. Questo significa che l'interpretazione potrebbe portare con sè dei parametri, che però saranno assenti in altre possibili istanze di Interpretation. Una volta ottenuta l'interpretazione del messaggio, si tratta di inviare una risposta all'utente, rappresentata nella nostra architettura dall'entità "BotMessage". Alcuni messaggi di risposta possono voler attendere una conferma, perché la corretta interpretazione del messaggio inviato originariamente dall'utente è critica e sbagliarla porterebbe ad inficiare i risultati del lavoro di interpretazione fatto. È per esempio il caso sempre dei messaggi di registrazione delle misurazioni di pressione, durante l'interpretazione dei quali i valori estratti potrebbero essere sbagliati e prima di salvarli nel *database* si chiede all'utente se i valori ottenuti sono corretti o meno. Il processo di gestione di un generico messaggio così come descritto è riassunto nel diagramma di interazione della figura 4.3. Questo diagramma permette anche di specificare meglio la differenza di comportamento tra "BotMessage" e "BotMessageWithConfirmation", dove quest'ultimo è un sottotipo del primo e rappresenta i messaggi inviati dal sistema che necessitano di attendere conferma. Questi non

solo possono essere inviati, come i BotMessage, ma possono far inviare anche un secondo testo che è la risposta alla conferma data dall'utente, suddividendo tra due messaggi quello che per noi logicamente è uno solo. Sarà quindi compito di "MessageDispatcher" tenere conto di quali sono gli utenti da cui si attende una conferma e quali no.

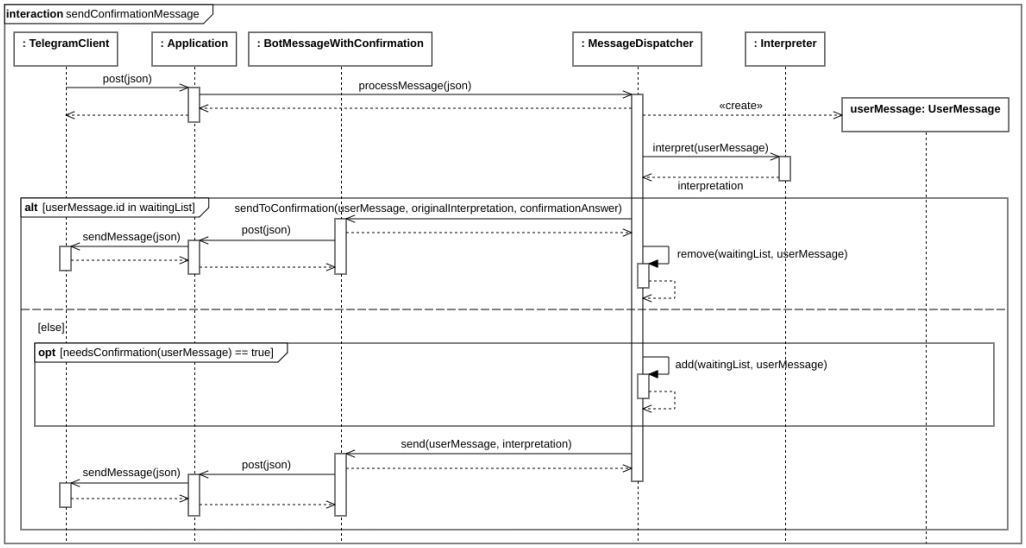

Figura 4.3: Diagramma di interazione che mostra la gestione di un messaggio da parte del sistema

Il database è gestito interamente dal componente "DBManager", come è evidente dal nome stesso. Questo gestisce poi a sua volta una gerarchia di entità tutta sua i cui elementi principali sono "User" e "Measurement". La prima rappresenta un utente come lo vede il *database*, quindi con tutti gli attributi necessari da memorizzare ed estrarre. User ha un attributo che permette di associare un utente nel sistema con il corrispondente nella piattaforma Telegram e tre valori booleani per controllare quale notifica è stata inviata, se è stata inviata, per ricordare all'utente di misurarsi la pressione. Infine, possiede la data di invio dell'ultima notifica per ricordare all'utente di prendere la propria terapia. Measurement invece rappresenta una misurazione con tutti i dati che le appartengono: l'utente che l'ha effettuata, il *timestamp* dell'istante in cui è stata presa e i valori contenuti in essa. Questo è un sottotipo di "UndatedMeasurement", quest'ultimo nato per distinguere la misurazione così come indicata poco fa da quella intesa come semplice raccolta di valori. UndatedMeasurement infatti possiede solamente gli attributi inerenti la pressione sistolica, la pressione diastolica e la frequenza cardiaca. Poiché questa gerarchia di entità

è stata poi utilizzata per costruire il *database*, è stato prodotto un diagramma *Entity-Relationship* che la contiene visibile nella figura 4.4.

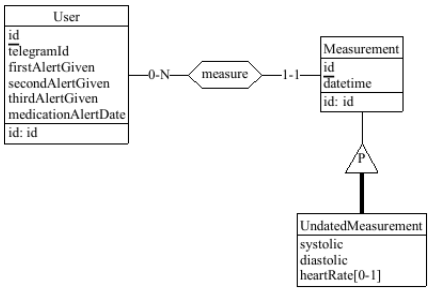

Figura 4.4: Diagramma *Entity-Relationship* che contiene la gerarchia di entità che compone il *database*

4.4 Tecnologie scelte per lo sviluppo

Nelle specifiche, la scelta della piattaforma Telegram come *front-end* per il *chatbot* è stata dettata a partire dalla necessità di produrre il sistema nel più breve tempo possibile. Infatti, in questo modo, non è stato necessario lo sviluppo di un'applicazione che permettesse l'interazione tra l'utente e il *bot* vero e proprio. Se invece fosse stata sviluppata, si sarebbe dovuto farlo per almeno due sistemi operativi, Android e iOS, che avrebbe non poco complicato la sua creazione. D'altronde, per raggiungere quanti più utenti possibile, è fondamentale che l'applicazione fosse stata multi-piattaforma. Inoltre, la scelta di Telegram ha permesso di avere già a disposizione una *user interface* semplice ed intuitiva per la totalità degli utilizzatori di *smartphone*, data la forte popolarità delle applicazioni di messaggistica istantanea in generale. In più, non si rende necessario nessun download o approvazione da parte di alcun *app store*, anche se chi utilizzerà il servizio dovrà avere Telegram e un account attivo su di esso.

È stata considerata anche la piattaforma di Facebook "Messenger" per lo sviluppo del sistema. Questa scelta si sarebbe però rivelata problematica su più aspetti, in primo luogo da un punto di vista puramente tecnico. Per poter sviluppare un *bot* di Messenger è necessario che il sistema abbia una propria pagina Facebook. Attraverso questa, esso sarebbe stato autorizzato all'utilizzo delle API di Facebook, venendo identificato come "app Facebook". In secondo luogo, l'uso di servizi legati al *social network* avrebbe potuto destare sospetti per quanto riguarda la *privacy* dei dati che vengono immessi nel sistema, che essendo dati sanitari sono considerati sensibili. Telegram per contro ha sempre

fatto della *privacy* tra utenti la sua bandiera ed inoltre tutte le sue applicazioni sono *open-source*.

Il *back-end* è stato sviluppato in Java 14, essendo un linguaggio molto flessibile. È inoltre molto diffuso, dato che è disponibile per la stragrande maggioranza di sistemi operativi *desktop*, e molto popolare tra gli sviluppatori, cosa che ha permesso facilmente di reperire librerie di qualsiasi tipo. Telegram indica Java tra i linguaggi per i quali è stata sviluppata una API capace di interfacciarsi con i suoi *bot* ed è stata utilizzata proprio una delle API che l'applicazione di messaggistica stessa consigliava. L'utilizzo del linguaggio Java ha permesso inoltre di utilizzare la libreria "VertX" per non dover costruire da zero l'architettura di controllo del *back-end*, demandandole la gestione del ciclo di esecuzione di base, nonché la creazione dei *web client* e dei *web server* necessari dal sistema sviluppato.

Il servizio utilizzato per effettuare "Natural Language Understanding", ovverosia nel nostro caso la comprensione dei messaggi testuali, è stato "Wit.ai" di Facebook. La scelta è ricaduta su di esso per la sua semplicità d'uso, essendo il *software* di addestramento dell'interprete di tipo WYSIWYG con un'interfaccia grafica che non richiedeva alcuno sforzo nel comprendere come utilizzarla. Wit.ai dà anche la possibilità di essere usato gratuitamente e permette di riconoscere le entità all'interno del testo scritto in linguaggio naturale, operazione di fondamentale importanza per estrarre i valori delle misurazioni di pressione. Per questo tipo di servizio sono state considerate alcune alternative come "Dialogflow" di Google e "Bot Framework" di Microsoft. Entrambe queste API però non sembrano offrire un livello di uso gratuito accessibile agli sviluppatori, perciò sono state scartate. Allo stesso modo è stato scartato "Watson" di IBM perché offre separatamente i servizi di Natural Language Classification, per la comprensione delle richieste dell'utente, e di Natural Language Understanding, per l'estrazione di entità dal testo. Infine, altri *framework* come "Pandorabots" o "Chatterbot" sono troppo complessi per l'uso che se ne doveva fare. Il primo richiedeva l'uso di un linguaggio di *markup* specifico per l'AI, mentre il secondo di fornire in anticipo dei possibili dialoghi tra l'utente e il *chatbot*, quando l'idea era di costruirlo su di un modello di conversazione a "botta e risposta".

Infine, per quanto riguarda il servizio utilizzato per effettuare la comprensione delle immagini, si è deciso di utilizzare "Tesseract". Esso è un *software* per l'*Optical Character Recognition*, capace perciò di riconoscere i caratteri presenti in un'immagine che rappresenta un testo scritto. Una fotografia di uno schermo però è molto differente da un foglio con delle parole, perciò è dovuta intervenire a monte una fase di *preprocessing* dell'immagine. Questa è stata portata avanti grazie all'ausilio della libreria per la manipolazione delle immagini "OpenCV". È stato scritto prima uno *script* in Python capace,

anche se non sempre, di ottenere un'immagine in bianco e nero che contiene solamente le cifre da riconoscere, poi lo *script* così creato è stato portato nel linguaggio Java.

4.5 Progettazione di dettaglio e sviluppo

Per quanto riguarda l'implementazione dell'architettura descritta precedentemente, non è necessario aggiungere molto di più rispetto a quanto già detto. Tutte le entità sono state tradotte in interfacce e sono state associate nel codice ad una classe che è la loro implementazione. Fanno eccezione poche di esse, ad esempio "Interpreter" è implementata sia da "NaturalLanguageInterpreter" che da "ImageInterpreter", come ci si poteva aspettare sulla base di quanto detto precedentemente. Un'altra gerarchia di classi importante è quella che discende da "BotMessage". Si è deciso infatti di costruire una classe per ogni tipo di risposta che il *chatbot* può dare alle richieste dell'utente. Questo ha portato ad avere ben sei sottoclassi, ognuna delle quali condivideva parte del proprio codice con le altre, dato che tutte hanno compiti simili. Si è perciò costruita una classe astratta, "AbstractBotMessage", col compito di fattorizzare i comportamenti comuni racchiudendoli all'interno di metodi *protected*, offerti quindi per l'uso alle sottoclassi. Questo *pattern* è conosciuto con il nome di "Subclass Sandbox" [12] e, a patto di mantenere basso il numero di metodi della classe astratta, è pensato proprio per fattorizzare il comportamento delle sottoclassi. La struttura di questa gerarchia è visibile nella figura 4.5, assieme alle relazioni che queste classi hanno con le interfacce "DBManager".

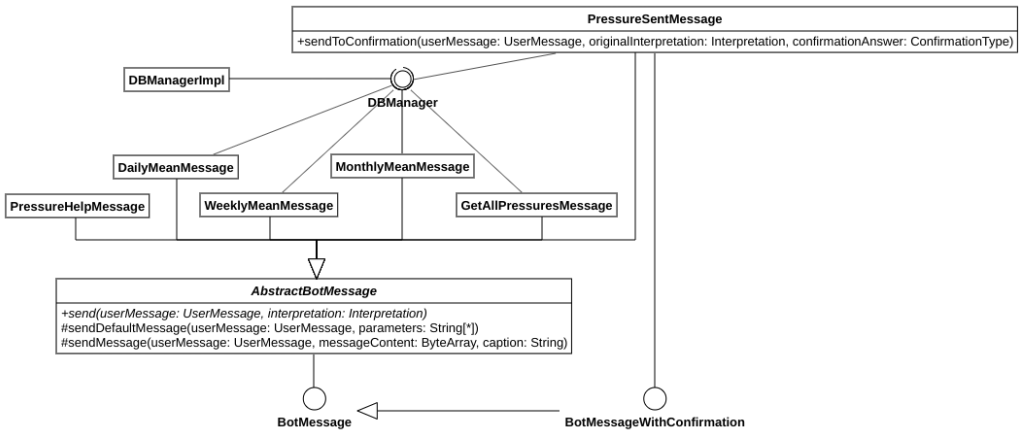

Figura 4.5: Diagramma delle classi che rappresenta la gerarchia di elementi che discende da "BotMessage" e le relazioni di questi ultimi con "DBManager"

Un'altra classe interessante è l'implementazione di "Interpretation", "InterpretationImpl". Essa è costruita a partire dai risultati che il servizio di "Natural Language Understanding" ci fornisce, il quale viene contattato attraverso delle *request* HTTPS le quali producono *response* il cui *body* è formattato in JSON. Poiché la costruzione delle istanze di questa classe avviene per gradi, mano a mano che viene effettuato il *parsing* del JSON, si è reso necessario salvare i valori intermedi estratti da esso da qualche parte. Piuttosto che inserire dei *setter* ad InterpretationImpl, si è preferito aggiungere metodi simili in una classe che avesse lo scopo specifico di costruire nuove istanze della precedente. Questo ha reso le istanze di InterpretationImpl immutabili e perciò più sicure nel loro utilizzo, nonché *thread-safe*. Da qui è nata la necessità di avere un *builder* per questa classe. Inoltre il *parsing* del JSON potrebbe non andare a buon fine, perché dei valori che erano attesi non sono presenti o si è verificato qualche altro errore di sorta. Anziché perciò lanciare un'eccezione o fare in modo che il metodo "build" del *builder* restituisca `null`, si è deciso di utilizzare il cosiddetto "Null Object Pattern" [9]. Questo *pattern* ha reso più robusto il codice in quanto è così contemplata da esso stesso la possibilità di errore ed è facile controllare quando questo si manifesta senza alterare il normale flusso di esecuzione del *software*.

L'ultima interfaccia di cui è interessante analizzare l'implementazione è "Application". È importante notare infatti che la classe "ApplicationImpl" necessita di un oggetto di tipo "Configuration" per poter funzionare. Questo contiene parte delle proprietà necessarie per poter appunto configurare i vari componenti che fanno parte dell'applicazione, in particolar modo i parametri "sensibili". Con questo termine si indicano tutti quei parametri che permettono l'accesso ai servizi utilizzati dal sistema a suo nome o alle componenti del sistema stesso, come ad esempio il *database*. Questi parametri sono perciò *token* di identificazione, nomi utente, password e simili che non dovrebbero essere condivisi con nessuno e quindi a maggior ragione non dovrebbero essere tracciati in una *repository* pubblica quale è quella che contiene il codice del sistema. Per fare questo, l'implementazione dell'interfaccia Configuration è stata pensata per essere de-serializzata a partire da un file di configurazione, così da non cablare questi valori nel codice. Il file è formattato in JSON, un formato aperto, che, benché più complesso da de-serializzare, permette di inserire nuovi parametri o rimuoverli dal file in maniera flessibile, in caso di successive espansioni del sistema, con minime modifiche di codice. Inoltre, dato che il file non è chiaramente tracciato nella *repository*, in caso di installazione del sistema su di una nuova piattaforma, il suo formato aperto permette una facile ricostruzione del file stesso con nuovi valori o con quelli precedentemente presenti.

Come detto in precedenza, Application rappresenta il gestore del flusso di

controllo del *chatbot*. Per questo motivo, sarà anche quel componente capace di mandare in esecuzione i *task* su richiesta degli altri componenti qualora si rendesse necessario. In questo sistema gli unici compiti che vengono eseguiti sono periodici e perciò l'interfaccia Application permette unicamente la creazione di questi. I task periodici sono usati per l'invio dei messaggi di notifica che sono inclusi nelle specifiche, i quali vanno inviati con intervalli di tempo tra gli uni e gli altri esprimibili in giorni. Per poter portare a termine l'invio di ciascuno di essi nelle giuste tempistiche, è bastato quindi realizzare un singolo task che ha periodo pari ad un giorno e che quindi giornalmente controlla se si sono verificate le condizioni per effettuarlo. Qui di seguito nella figura 4.6 si mostra il diagramma degli stati che il sistema implementa per l'invio delle notifiche seguendo le specifiche date. Come è facile intuire da quanto detto, si tratta di una macchina a stati finiti sincrona.

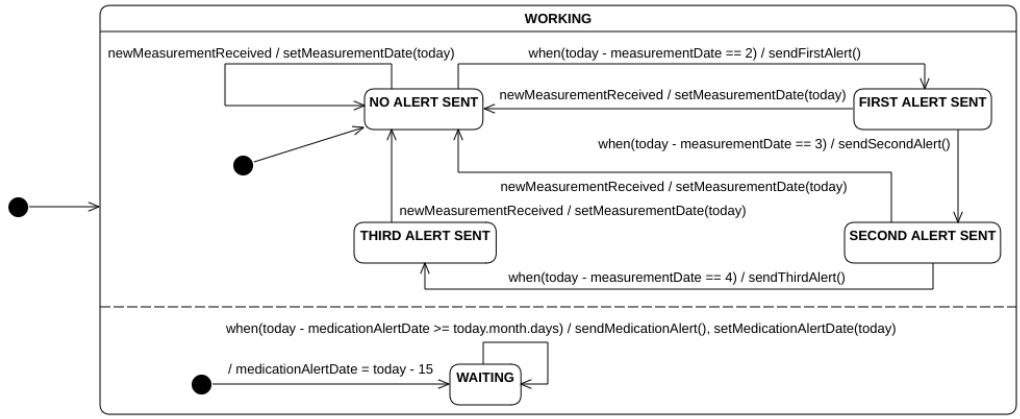

Figura 4.6: Diagramma degli stati del sistema per l'invio delle notifiche

Per quanto riguarda il *database*, dal diagramma E/R precedentemente illustrato è stato costruito lo schema logico presente nella figura 4.7.

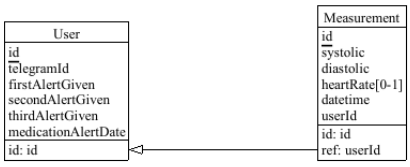

Figura 4.7: Schema logico del *database* costruito

Come si può vedere è stato operato un collasso verso il basso della gerarchia. Questo perché i due tipi di misurazioni differiscono soltanto per gli attributi

che contengono, non esistono infatti istanze di "UndatedMeasurement" nel *database* che non siano anche istanze di "Measurement". Perciò, se si vuole estrarre da un'istanza di Measurement una di UndatedMeasurement, basterà semplicemente applicare la corretta operazione di proiezione.

4.6 Alpha-testing e previsione di deployment

L'*alpha-testing* di questo sistema è stato effettuato manualmente, senza l'ausilio di alcuno strumento automatico la creazione e l'esecuzione di *unit-test*. Questo per due motivi: il primo è legato alla natura estremamente reattiva del sistema, che eseguendo le operazioni in *callback* innestate non ha un flusso di esecuzione lineare da poter testare in semplicità. In secondo luogo, il sistema è molto piccolo, dato che le sue specifiche sono poche e la loro implementazione è stata mantenuta semplice. Perciò si sono considerati i test manuali come sufficienti per effettuare la copertura di tutti i casi d'uso del *chatbot*, anche quelli limite.

Il sistema non ha ancora ricevuto un *deployment* ufficiale, essendo in uno stadio prototipale. Proprio per questo motivo, si è cercato di adottare quante più strategie possibili per non vincolare ad uno specifico ambiente *software* l'installazione del sistema. Dal linguaggio di programmazione alle librerie e ai servizi utilizzati, tutto è multi-piattaforma, in modo da poter scegliere le combinazioni di sistema operativo ed *hardware* sottostante più adeguate. Chiaramente esistono comunque dei vincoli: sul server in cui sarà installato il sistema dovrà essere possibile installare una Java Virtual Machine capace di supportare Java 14, anche se probabilmente si può scegliere una versione più retrodatata del linguaggio come la 10, il software per l'OCR Tesseract e la libreria OpenCV. Il fatto di non avere ancora raggiunto la fase di *deployment* significa anche che il test con i pazienti non è ancora stato fatto. Questo però non significa che il suo funzionamento non abbia ricevuto una valutazione, anzi. Il sistema è stato mostrato al committente, il dottor Martino Pengo, che si è detto soddisfatto del lavoro svolto. È stato giudicato conforme a tutte le specifiche date, non solo nella parte più dedicata al medico, ma anche in quella relativa all'interazione con il paziente. Questo giudizio è stato poi confermato dal dottor professor Grzegorz Bilo dell'Università degli Studi di Milano - Bicocca, invitato dal dottor Pengo stesso a valutare il funzionamento del sistema.

Si mostrano quindi alcune catture di schermata col fine di esemplificare il funzionamento del sistema realizzato così com'era nel momento della sua approvazione. Nella figura 4.8 si può vedere che cosa succede all'inizio di qualsiasi conversazione con il *chatbot*. Prima di tutto viene mostrato un messaggio

che riassume in poche righe le capacità del *bot*. Questo non viene inviato dal *chatbot*, ma mostrato dalla piattaforma Telegram stessa, prima che l'utente lo avvii con il comando "\start". Il secondo messaggio inviato dal *chatbot* è invece la risposta che dà al suo avvio, cercando di incoraggiare l'utente ad esplorarlo sfruttando le funzionalità che esso possiede.

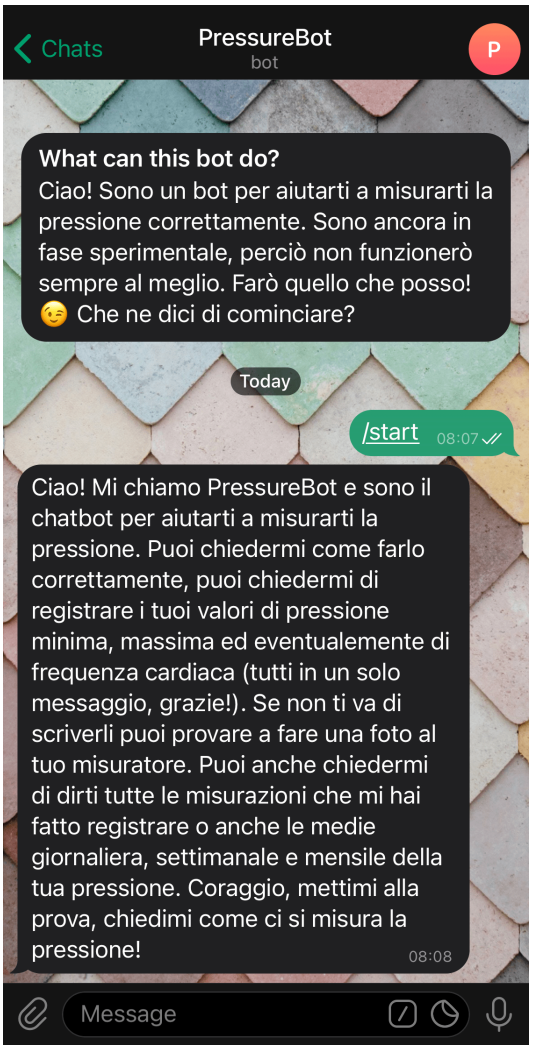

Figura 4.8: *Screenshot* dei messaggi iniziale e di avvio del *chatbot* così come sono visibili su Telegram

Nella figura 4.9 si vede che il messaggio successivo inviato dall'utente è per ottenere più informazioni sul come misurarsi correttamente la pressione. La domanda è stata posta in quello che, ovviamente, è soltanto uno dei possibili modi con cui si può effettuare questa richiesta al *bot*. Altre domande o anche

altre affermazioni avrebbero potuto portare alla risposta da parte del sistema che è possibile vedere, purché esprimessero abbastanza esplicitamente la volontà di sapere come misurarsi correttamente la pressione.

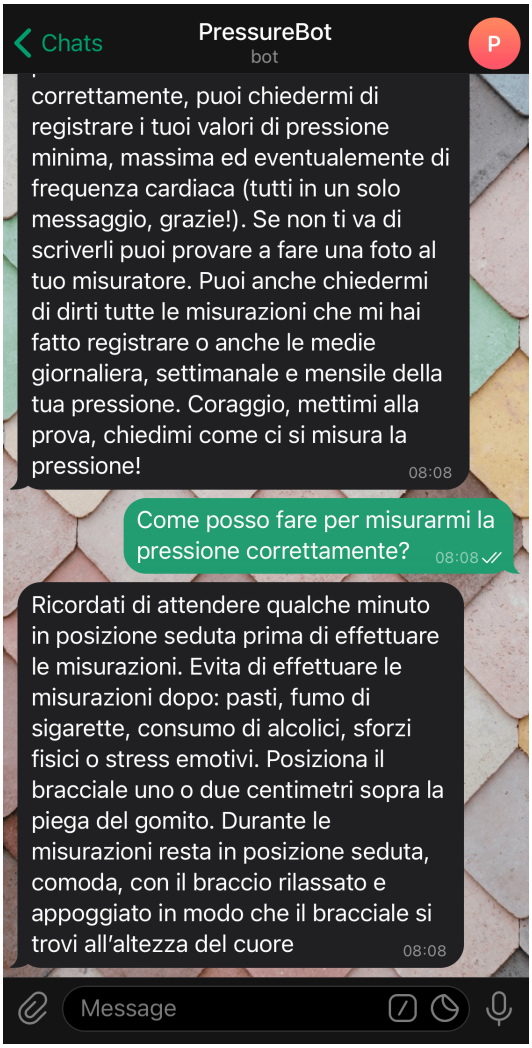

Figura 4.9: *Screenshot* del messaggio di risposta alla richiesta da parte dell'utente di voler sapere come potersi misurare correttamente la pressione

Nella figura 4.10 invece viene mostrata la fase iniziale di uno scambio di messaggi per la registrazione di una misurazione di pressione. Come detto in precedenza, è fondamentale che il *chatbot* non sbagli nell'interpretazione del messaggio, pena la registrazione e la successiva analisi da parte del medico di valori pressori sbagliati. Per questo motivo viene mostrata all'utente una tastiera su cui sono presenti solamente i pulsanti "sì" e "no" per confermare

che quelli che il sistema è riuscito a leggere sono i valori corretti. Può accadere che il sistema riesca a riconoscere la volontà dell'utente di registrare una misurazione, ma il formato del messaggio sia sbagliato, perché ad esempio manca un valore necessario. In tal caso verrà mostrata una notifica di errore che spiegherà come scrivere un messaggio di richiesta per la registrazione di una misurazione in modo che sia comprensibile dal *chatbot*.

Figura 4.10: *Screenshot* della richiesta di registrazione di una misurazione di pressione e della relativa risposta, la quale mostra all'utente una tastiera temporanea per rispondere al messaggio

A questo punto, per l'utente, il sistema è "bloccato" in questo stato. Ciò significa che finché l'utente non confermerà o negherà la correttezza della misurazione, il *chatbot* pretenderà una risposta, invitandolo a ripeterla qualora

non avesse capito. Questo perché la comparsa della tastiera sostitutiva non implica l'impossibilità dell'uso di quella QWERTY standard. Questo significa che un utente potrebbe voler oltrepassare la tastiera semplificata fornitagli e rispondere con un messaggio personalizzato alla richiesta di conferma della misurazione. Data l'esistenza di questa possibilità, il *chatbot* è stato addestrato a riconoscere anche questo tipo di messaggi, come si può vedere nella figura 4.11. In questa stessa figura si può vedere che il *bot* è capace di interpretare correttamente anche le misurazioni che contengono la frequenza cardiaca oltre

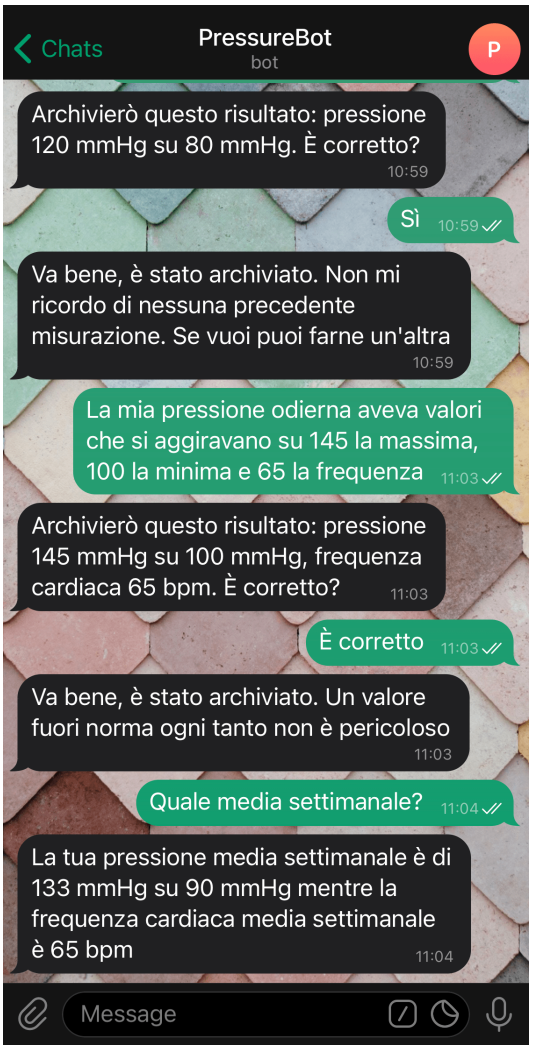

Figura 4.11: *Screenshot* che mostra la conferma dell'utente all'interpretazione della misurazione di pressione, una nuova registrazione di pressione che contiene anche la frequenza cardiaca e una richiesta di ottenere la media settimanale dei valori registrati

che la pressione sistolica e la pressione diastolica. Infine, si può vedere la risposta che il sistema dà alla richiesta della media settimanale dei valori.

Da ultimo, nella figura 4.12, si mostra la richiesta di tutte le misurazioni precedentemente registrate.

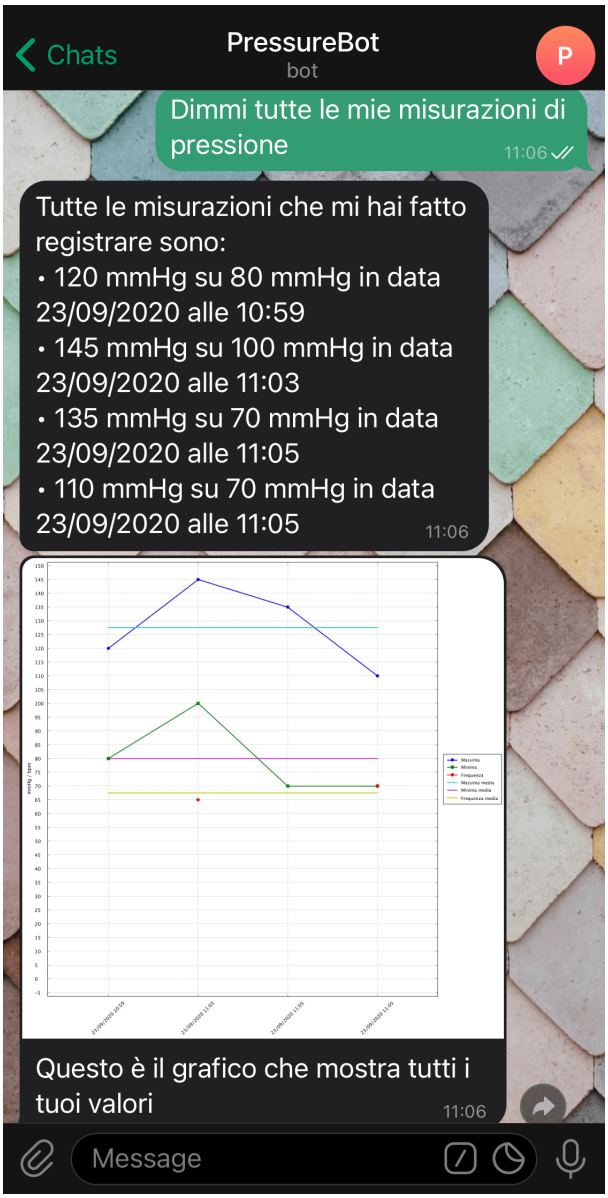

Figura 4.12: *Screenshot* della richiesta di tutte le misurazioni precedentemente registrate e delle relative risposte

Come risposta vengono inviati due messaggi: il primo contiene un riepilogo con tutte le misurazioni e i relativi *timestamp* in cui sono state fatte. Il secondo

contiene invece il grafico, con le stesse misurazioni del messaggio precedente, che mostra l'andamento dei valori pressori e della frequenza cardiaca, oltre alla media di ciascuno dei parametri calcolata su tutti i valori mostrati. Nella figura 4.13 è possibile vedere meglio il grafico che compare nella 4.12. In esso, lungo le ascisse sono posti i *timestamp* delle misurazioni effettuate e lungo le ordinate i valori contenuti nella misurazione per ciascun parametro.

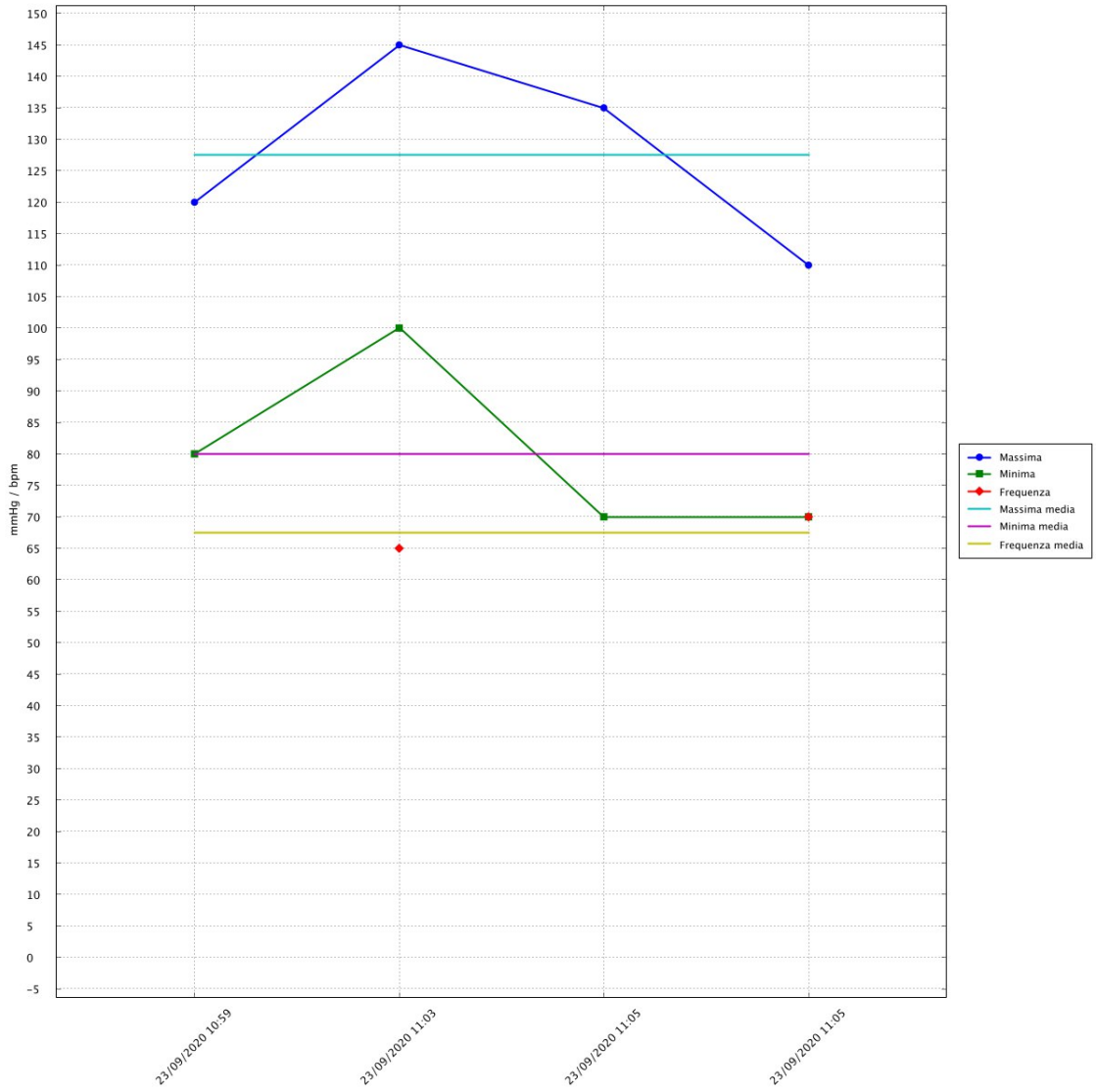

Figura 4.13: Grafico ingrandito già presente nella precedentemente figura con i valori delle misurazioni di pressione e di frequenza cardiaca e le medie per ciascuno di questi tre parametri

Conclusioni

Per quanto riguarda l'analisi dei dati, i risultati purtroppo sono stati mediocri. Molte delle cose che sono state evidenziate attraverso le variabili dei modelli e la loro importanza erano già note da tempo in campo medico. Per questo non è stata raggiunta nessuna nuova conoscenza capace di contribuire a modifiche delle correnti linee guida sull'ipertensione. Come per tutte le ricerche, però, anche questo è un risultato positivo: significa che i dati così come sono non sono in grado di effettuare delle predizioni interessanti, anche se queste non erano l'obiettivo finale del lavoro fatto. Esistono quindi ancora aspetti che si possono migliorare. Si potrebbe per esempio indagare il metodo di costruzione del *dataset*, per vedere se e come migliorare ulteriormente la qualità dei dati e come cambiare il metodo di raccolta degli stessi in modo coerente con quanto trovato. Si può anche condurre l'analisi in altri modi. Ad esempio indagando anche i pazienti che sono noti ipertesi, indipendentemente dal fatto che prendano farmaci per l'ipertensione o meno. Possono essere presi separatamente per vedere quanto i valori delle misurazioni effettuate durante il questionario sono dei buoni predittori, oppure possono essere inclusi nell'analisi fatta, ma indicando in corrispondenza delle loro istanze la variabile da predire come automaticamente positiva. Si può poi pensare di effettuare il *deployment* di questo modello in un qualche *software*, automatizzando il lavoro di raccolta dei questionari. Questo migliorerebbe la qualità dei dati raccolti e fornirebbe alle persone un riscontro diretto del proprio stato di salute a partire dalle proprie risposte.

Osservando invece il lavoro effettuato per lo sviluppo del *chatbot*, i risultati sono stati molto più soddisfacenti. Anche se il vero test con i pazienti deve essere ancora effettuato, la dimostrazione del sistema al committente ha riscosso apprezzamenti nel suo interagire con l'utente, oltre che nelle funzionalità pensate anche per il medico. Il sistema è stato sviluppato rapidamente, come se fosse una *spike solution*, ma in realtà la qualità del codice è rimasta alta. Questo perché si è tenuto conto dei futuri sviluppatori che dovranno prendere in mano il progetto e migliorarlo, o quanto meno aggiungere le *feature* che si renderanno mano a mano necessarie. Il sistema infatti può essere espanso in più modi. Si potrebbe renderlo indipendente dalla piattaforma Te-

legram, sviluppando più applicazioni native che facciano da *front-end*. Questo significherebbe anche avere accesso ai sensori del dispositivo da cui l'utente si sta interfacciando al *chatbot*, che lo potrebbe arricchire di possibili nuove funzionalità. Si potrebbero anche cambiare i servizi utilizzati per il "Natural Language Understanding" e per lo "Optical Character Recognition", cioè "Wit.ai" e "Tesseract" rispettivamente. Li si potrebbe sostituire con servizi più complessi e più avanzati, specialmente per quanto riguarda l'OCR. I risultati di riconoscimento delle cifre nelle immagini sono scarsi, anche a causa di un *preprocessing* delle foto che non è costruito su solide basi. Per migliorare quest'ultimo si potrebbero utilizzare nozioni di visione artificiale più approfondite per ottenere un tasso di successo più alto.

Ringraziamenti

Si ringraziano innanzitutto i professori Alessandro Ricci e Sara Montagna, senza i quali questa tesi non sarebbe stata possibile, per i consigli, l'aiuto dato e per avermi seguito nel percorso che ho fatto. Si ringrazia anche il dottor Martino Pengo per averci portato a conoscenza dei problemi e degli obiettivi che sono stati discussi e per avere fornito consigli su come portare a termine i due progetti che hanno costituito questa tesi. Inoltre, lo si ringrazia assieme al professor Grzegorz Bilo per i suggerimenti sul come migliorare l'interfaccia utente del *chatbot* sviluppato. Si ringrazia infine l'ingegner Andrea Faini per la costruzione del *dataset* sul quale è stata effettuata l'analisi dei dati.

Bibliografia

[1] Kevin Ashton. That "Internet of Things" Thing. *RFID Journal*, 2009.

[2] BBC. Horizon, season 1 episode 6: "The knowledge explosion". BBC Television, 1964.

[3] Hans Berliner. Herbert A. Simon (1916-2001): A Life's Appraisal. *ICGA Journal*, 24(1), 2001.

[4] Timothy Chou. *Precision - Principles, Practices and Solutions for the Internet of Things*. McGraw Hill Education (India), 2017.

[5] Google Danmark. [Think with Google] ThinkDK 2018: Daniel Hulme, AI and the future of business. Youtube, 2018.

[6] Andreas Kaplan e Michael Haenlein. Siri, Siri, in my hand: Who's the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence. *Business Horizons*, 62, 2019.

[7] Justin B. Echouffo-Tcheugui et al. Risk Models to Predict Hypertension: A Systematic Review. *PlosOne*, 8(7), 2013.

[8] Paul K. Whelton et al. 2017 ACC/AHA/AAPA/ABC/ACPM/AGS/APhA/ASH/ASPC/NMA/PCNA Guideline for the Prevention, Detection, Evaluation, and Management of High Blood Pressure in Adults: A Report of the American College of Cardiology/American Heart Association Task Force on Clinical Practice Guidelines. *Hypertension*, 71(6), 2018.

[9] Martin Fowler. *Refactoring: Improving the Design of Existing Code*. Addison-Wesley, 1999.

[10] Istat. *Annuario statistico italiano 2019*. Istat, 2019.

[11] Tom M. Mitchell. *Machine learning*. McGraw Hill, 1993.

[12] Robert Nystrom. *Game Programming Patterns*. Genever Benning, 2014.

[13] David Rose. *Enchanted Objects: Innovations, Design and the Future of Technology*. Scribner, 2015.

[14] John R. Searle. Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3), 1980.

[15] Eliza Strickland. IBM Watson, heal thyself: How IBM overpromised and underdelivered on AI health care. *IEEE Spectrum*, 56(4):24–31, 2019.

[16] Eric J. Topol. High-performance medicine: the convergence of human and artificial intelligence. *Nature medicine*, 25, 2019.

[17] Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236), 1950.

[18] Peter Waher. *Learning Internet of Things*. Packt Publishing Ltd., 2015.