ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
UNIVERSITY OF PIRAEUS

DEMOKRITOS

Chatbots in Healthcare: Towards AI-enabled general diagnosis and medical support

by

Vasiliki Vryoni

Submitted
in partial fulfillment of the requirements for the degree of Master of
Artificial Intelligence
at the
UNIVERSITY OF PIRAEUS

Athens, July 2021

Chatbots in Healthcare: Towards AI-enabled general diagnosis and medical support

Vasiliki Vryoni

MSc. Thesis, MSc. Programme in Artificial Intelligence

University of Piraeus, NCSR "Demokritos", July 2021

Author: Vasiliki Vryoni

II-MSc “Artificial Intelligence”

Athens, July 2021

Approved by the examination committee

(Signature)	(Signature)	(Signature)
. . . .	. . . .	. . . .
Ilias Maglogiannis Professor	Georgios Vouros Professor	Michail Filippakis Associate Professor

Chatbots in Healthcare: Towards AI-enabled general diagnosis and medical support

by

Vasiliki Vryoni

Submitted to the II-MSc "Artificial Intelligence" on July 2021
in partial fulfillment of the
requirements for the MSc degree

Acknowledgments

First of all, I would like to sincerely thank my Supervisor Professor Ilias Maglogiannis of the Department of Digital Systems of the University of Piraeus, for the cooperation, advice and guidance through this challenging journey.

I also wish to express my deepest gratitude to my family for supporting me during my studies. This thesis is dedicated to them, as it would not have been completed without their love and encouragement.

Last but not least, very special thanks to my friends Haris, Marialena, Panagiotis M. and Panagiotis T. for their unconditional support and understanding during these two intense academic years.

Περίληψη

Οι Πράκτορες Συζήτησης, αλλιώς γνωστοί ως ψηφιακοί βοηθοί, είναι λογισμικό Τεχνητής Νοημοσύνης σχεδιασμένο με σκοπό την πραγματοποίηση ζωντανής συνομιλίας μεσω κειμένου ή ομιλίας με ένα χρήστη ώστε να παρέχει διάφορες μορφές υπηρεσίας. Στον τομέα της Υγείας, οι πράκτορες συζήτησης μπορούν να παρέχουν αμφίδρομη ανταλλαγή πληροφορίας μεταξύ ασθενών και ειδικών του τομέα, με σκοπό τη συμβουλευτική και τα αρχικά στάδια ιατρικής βοήθειας, μειώνοντας σημαντικά το οικονομικό, αλλά και το κόστος σε χρόνο για όλους τους εμπλεκόμενους. Στην παρούσα διπλωματική εργασία υλοποιήθηκαν δύο ξεχωριστοί πράκτορες συζήτησης για τον τομέα της Υγείας. Ο πράκτορας DrBot συλλέγει συμπτώματα και κάποια βασικά χαρακτηριστικά του χρήστη, με σκοπό να του παρέχει μια αρχική διάγνωση για πιθανές ασθένειες από τις οποίες μπορεί να πάσχει, σύμφωνα με τα συμπτώματα που του περιγράφηκαν. Αυτό επιτυγχάνεται με τη βοήθεια του Infermedica API. Ο πράκτορας πρόβλεψης ρίσκου στεφανιαίας νόσου επίσης συλλέγει χαρακτηριστικά του χρήστη, τόσο δημογραφικά όσο και κλινικά, με σκοπό να προβλέψει το ρίσκο να νοσήσει από τη συγκεκριμένη ασθένεια. Στην μελέτη παρουσιάζονται ο τρόπος υλοποίησης του κάθε πράκτορα και στη συνέχεια πειράματα και παρατηρήσεις σχετικά με την αποτελεσματικότητα και τη χρησιμότητά τους. Τέλος, εξάγονται συμπεράσματα και προτάσεις για μελλοντικές βελτιώσεις της υλοποίησης.

Λέξεις κλειδιά: Πράκτορες Συζήτησης, Τεχνητή Νοημοσύνη, Μηχανική Μάθηση, Τηλε-ιατρική, Υγεία

Abstract

Chatbots, also known as conversational agents or digital assistants, are artificial intelligence (AI) software applications used to conduct an online chat conversation via text or text-to-speech, in lieu of providing direct contact with a live human agent. In healthcare, chatbots can create bidirectional information exchange with patients, in order to effectively support patients and health professionals in therapeutic settings beyond on-site consultations by reducing the healthcare costs and improving accessibility to medical knowledge for everyone. In this master thesis, the multifunctional role of chatbots is examined, in two different use cases in healthcare domain. DrBot is designed to collect symptoms from users, conduct an interview and provide a potential diagnosis and triage consultation based on their answers. This is achieved with the support of Infermedica medical API. HeartRisk chatbot collects some target medical and lifestyle information from users, aiming to perform a prediction regarding a heart disease risk. Preliminary experiments about the effectiveness of the proposed approaches are reported and future work is suggested.

Keywords: Chatbots, AI, Machine Learning, Telemedicine, eHealth

Contents

List of Tables

List of Figures

List of Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
NLP	Natural Language Processing
EHR	Electronic Health Records
API	Application Programming Interface
JSON	JavaScript Object Notation
CVD	Cardiovascular Disease
CHD	Coronary Heart Disease
RTDB	Real-time Database
DT	Decision Trees
SVM	Support Vector Machines
KNN	K-Nearest Neighbours
ROC	Receiver Operating Characteristic
AUC	Area under the ROC Curve

Chapter 1

Introduction

This section consists of a brief introduction to chatbots and their importance in Telemedicine.

1.1 Dialogue Systems

Conversational AI has been defined as “the study of techniques for creating software agents that can engage in natural conversational interactions with humans” [1].While the idea of interacting with a computer using text or voice has been around for a long time, it is only recently that it has become a reality. Nowadays, people can talk to digital personal assistants on their smartphones, they can ask questions or issue commands to voice-enabled smart speakers, and they can navigate using voice-based systems in their cars. In other words, Conversational AI has become ubiquitous. Historically there have been five distinct traditions in dialogue systems research involving communities that have largely worked independently of one another [2]. These are:

- Text-based and Spoken Dialogue Systems.
- Voice User Interfaces.
- Chatbots.
- Embodied Conversational Agents.
- Social Robots and Situated Agents.

Our study is focused on Chatbots and specifically of those applied in Healthcare Domain.

1.2 Chatbots

A chatbot is an artificial intelligence (AI) software that can simulate a conversation (or a chat) with a user in natural language through messaging applications, websites, mobile apps or through the telephone. A conversational agent is often described as one of the most advanced and promising expressions of interaction between humans and machines. However, from a technological point of view, a chatbot only represents the natural evolution of a Question Answering system leveraging Natural Language Processing (NLP). They are often used for user-friendly customer-service triaging. Instead of having a conversation with another person, the user talks with a bot that is powered by basic rules or Machine Learning (ML).

Every chatbot serves a specific purpose; health chatbots are designed to help with health-related issues. Health chatbots could potentially provide many different services. They might give the user health-related information. They can help set up appointments and later send reminders for them. While they cannot make official diagnoses, if you tell them your symptoms, they can give you a likely diagnosis. In some cases, health chatbots are also able to connect patients with clinicians for diagnosis or treatment. The general idea is that in the future, these talking or texting smart algorithms might become the first contact point for primary care [3]. Patients will not get in touch with physicians or nurses or any medical professional with every one of their health questions but will turn to chatbots first. If the little medical helper cannot comfortably respond to the raised issues, it will transfer the case to a real-life doctor.

1.3 Healthcare Chatbots concerns

Chatbots provide instant conversational responses and make connecting simple for patients. And when implemented properly, they can help care providers to surpass patient expectations and improve patient outcomes. However, AI solutions sometimes lack the most important quality to good care delivery: a human touch. Digital health platforms, especially platforms targeting mental health issues have seen an important growth nowadays, as in-person appointments have been relegated to the digital sphere. For those who cannot access therapy from a human clinician, mental health chatbot platforms are an increasingly popular digital alternative. Critics of these platforms have repeatedly questioned their efficacy, due to the lack of face-to-face connection and empathy between patient and clinician. Due to technologies like AI, ML, and NLP, it is said that chatbots have reached a level where they can gauge human sentiments. The uniqueness in every individual's behavior can confuse chatbots. Some people may prefer a casual talk, others may like the conversation to be formal.

Lack of empathy and unawareness of context can prove to be a very important obstacle in the healthcare space. In the following section, we have performed a literature survey on existing Healthcare Chatbot applications and have tried to address the aforementioned, as well as other possible issues.

Chapter 2

Related work

2.1 Literature Survey

As part of this study, a deep literature investigation has been performed as far as the medical chatbots and related eHealth applications are concerned. We have identified the research questions related to this field, the common areas of health applications so far, as well as concerns and suggestions for future improvements. This information has been summarized and is presented in the below table:

Table 2.1: Review and Results of recent studies

Reference	Objective	Experiment and Results	Remarks
[4]	Provide an overview of the features of chatbots used by individuals for their mental health as reported in the empirical literature.	53 unique studies, assessment of 41 different chatbots. Most common use of chatbots for: delivery of therapy, training, and screening. Inconsistency on how chatbots are measured.	Standardize measures and outcomes. Acceptability. Chatbot empathy.
[5]	Survey the landscape of health chatbots along three research questions: What illnesses are chatbots tackling? What patient competences are chatbots aimed at? Which chatbot technical enablers are of most interest in the health domain?	30 Articles related to health chatbots from 2014 to 2018. Main tackled illnesses: nutritional and neurological disorders. Patient competences: affect, in order to attain change behavior and personalization. Technical enablers: consumability.	Development for stakeholders other than patients. Improve NLP skills. Not only mobile app but also web.

Table 2.1: Review and Results of recent studies

Reference	Objective	Experiment and Results	Remarks
[6]	Review the current applications, gaps, and challenges in the literature on conversational agents in healthcare and provide recommendations for their future research, design, and application.	47 study reports (45 articles and 2 ongoing clinical trials). Most reported conversational agent applications in the literature were for treatment and monitoring, healthcare service support, and patient education, largely delivered via smartphone apps.	Predominance of small cases. Other platforms than smartphone apps.Lack of established method of evaluation.
[7]	Explore participants' willingness to engage with AI-lead health chatbots.	29 Participants Aged 18-22 years were interviewed, 215 users completed the online survey. Chatbot is convenient and anonymous, however the concerns about accuracy, trustworthiness and privacy, as well as the perceived lack of empathy are likely to compromise their adoption.	Concerns about the accuracy and security, interaction. Health chatbots should be a supplementary service rather than a replacement.

Table 2.1: Review and Results of recent studies

Reference	Objective	Experiment and Results	Remarks
[8]	Investigate the effectiveness of novel human-machine interaction paradigms for eHealth applications. Implemented: HOLMeS application, which provides medical recommendations for disease prevention.	Dataset: 13 diseases, 16733 patients. The chatbot was able to overcome the limitation of classical human-machine interaction, thus removing bias and allowing the patient to a freer and natural communication. Prevention pathway assessment: Percentge of 74.65 Area Under ROC Curve.	-

Table 2.1: Review and Results of recent studies

Reference	Objective	Experiment and Results	Remarks
[9]	Engaging adolescents along the transition journey from pediatric to adult care, using smartphones, text messaging and social media.	13 teenagers with a chronic medical condition. Mean engagement was 97 percent during the study period, rate of one message per week was preferred 3 areas: tracking medications, completing medication refills, and contacting a provider's office with questions, digital formats are appealing to adolescents and young adults.	More education on health insurance. Test the chatbot in larger populations.
[10]	Create a medical chatbot that can diagnose the disease and provide basic details about the disease before consulting a doctor.	-	Adding more combination of words. Voice conversation. Add support for more medical features.

Table 2.1: Review and Results of recent studies

Reference	Objective	Experiment and Results	Remarks
[11]	Follow up on older patients with cancer receiving chemotherapy at home.	Evaluation from 58 potential users (all of them 65 years or older), 8 benefited of the chatbot. The most valuable benefits were treatment management (40percent) and moral support. Feasibility to transform from conventional phone monitoring to monitor.	Test to a larger number of patients.
[12]	Keep track of diets, activity as well as weight, which is deemed more accurate than relying on user's self-report measure, for the sake of weight management.	Users can talk to the health chatbot, get the information realtime or take a bot's plans advice such as diet and exercise.	-

Table 2.1: Review and Results of recent studies

Reference	Objective	Experiment and Results	Remarks
[13]	Address different facets of behavioral health, deliver customized integrative support, psychoeducation, and interventions through brief conversations via existing Communication channels.	23 adolescent patients, coping with weight management and prediabetes symptoms. Positive progress towards their goals and targeted behaviors. 81 percent of the time, teen patients felt more comfortable sharing.	Development to engage other members involved in treatment. Use in an earlier stage of treatment, even in prevention.

Table 2.1: Review and Results of recent studies

The main results of the analysis performed can be divided in three main categories. The first category refers to the most common areas of health chatbot applications which include treatment and monitoring, education, lifestyle and behavioral change and diagnosis.

The second category concerns the main experimental results of health chatbot implementations. These results can be summarized to comfort of patients sharing private information, preference of chatbots due to anonymity, convenience and ease of use, conversation history logging, online availability of patient's medical file and popularity among adolescents and young adults, due to their familiarity with technology.

As for the last category, it covers the concerns raised so far and the future improvement areas. The concerns include the lack of familiarity with Chatbots since it is a recently thriving technology, the uncertainty about the quality of outcomes, the miscommunication and lack of trust, as well as the lack of a standardized method to test the results of a chatbot applications in Health Domain. In the future, Chatbot implementations should involve more AI and Machine Learning technologies rather than just storing patient information and providing advice and should be further implemented and studied in large-scale datasets, which is not so far commonly practiced.

2.2 Past Experience on Chatbot implementations in Health domain

2.2.1 Health Data Analytics

Healthcare AI-assisted products that are being nowadays developed are becoming more popular and are commonly used in clinics, hospitals, or homes. Some of the most interesting solutions offered by these businesses include AI-powered health assessment, early diseases detection using machine learning or AI-powered wearable devices to assist people with various disabilities. To develop such AI-assisted products, it is necessary to have a lot of high-quality data relevant to the problem that we want to solve in the healthcare market. Modern electronic health records

(EHRs) provide data to answer clinically meaningful questions. The growing data in EHRs makes healthcare ripe for the use of machine learning. However, learning in a clinical setting presents unique challenges that complicate the use of common machine learning methodologies. For example, diseases in EHRs are poorly labeled, conditions can encompass multiple underlying endotypes, and healthy individuals are underrepresented. [14]

2.2.2 Chatbot Implementations

A flurry of use cases — many of which center around customer (i.e. patient) service activities — have the potential to be overtaken by chatbots. Manual collection of data (e.g. medical history or current medications), for example, which is typically done over the phone or in the doctor's office prior, can be done automatically via chatbot. This means that patients will not be stuck on hold or spend longer than they need to at their appointments [5]. Through a simple chat window, chatbots provide patients the opportunity to:

- Find a doctor and make an appointment (https://kore.ai/about-kore/)
- Triage patients based on symptoms and symptom severity (https://symptomate.com/)
- Receive assistance with procedures recommended by their doctor (https://www.sensely.com/)
- Determine the dosage of a particular medicine and get reminders on when it is time to take a medicine (https://chatbottle.co/bots/florence-1)
- Establish a repository of their historical health data (https://chatbottle.co/bots/fitmeal-for-messenger)
- Get answers to their questions about medical equipment, lab procedures etc. (https://www.lumahealth.io/)

2.3 The shift to Telemedicine

2.3.1 Introduction

The current COVID pandemic has caused a lot of stress in the healthcare sector, with hospitals crowded with COVID-19 patients and handling regular consults. This has made medical chatbots very attractive, helping in scheduling appointments, customer support, symptom checks, providing nutrition and wellness information, mental therapy, etc. Chatbots are a significant part of the digital transformation in the healthcare sector and their reputation is day by day increasing.

Lockdowns and social distancing due to COVID-19 gave a significant boost to digital business models. Organizations had to find ways to keep up the operations, make business continuity plans, and engage the workforce working remotely. Many hospitals had been trying to implement telemedicine over the last years and COVID-19 gave that extra push.

Another tendency that people have these days is to search for information on Google for self-diagnosis, which is mostly missleading. Therefore, many people are turning towards healthcare chatbots for medical information.

2.3.2 Medical diagnosis

Medical diagnosis is the process of determining which disease or condition explains a person's symptoms and signs. It is most often referred to as diagnosis with the medical context being implicit. The information required for diagnosis is typically collected from a history and physical examination of the person seeking medical care. Often, one or more diagnostic procedures, such as medical tests, are also done during the process. Sometimes posthumous diagnosis is considered a kind of medical diagnosis. [15]

There are several challenges related to this process. Most important are the uncertainties related to observations and symptoms: rarely can the presence or absence of a single observation or symptom lead to a diagnosis, especially in the initial stages.

The second aspect is that observations and symptoms are linked to multiple illnesses, and often the presence or absence of a symptom does not directly indicate or exclude a given illness. In order to develop a reliable model for medical diagnosis, one must account for these two aspects. Other aspects that should be accounted for are the prevalence of the diseases and risk factors such as age and sex that influence both the structure of dependencies between symptoms and illnesses and the related likelihoods [16].

2.3.3 Cardiovascular Diseases

Cardiovascular diseases (CVD's) are the number one cause of death globally, taking an estimated 17.9 million lives each year. CVDs are a group of disorders of the heart and blood vessels and include coronary heart disease (CHD), cerebrovascular disease, rheumatic heart disease and other conditions. Four out of five CVD deaths are due to heart attacks and strokes, and one third of these deaths occur prematurely in people under 70 years of age. Individuals at risk of CVD may demonstrate raised blood pressure, glucose, and lipids as well as overweight and obesity. These can all be easily measured in primary care facilities. Identifying those at highest risk of CVDs and ensuring they receive appropriate treatment can prevent premature deaths [17]. Coronary heart disease in particular, continues to be a leading cause of morbidity and mortality among adults in Europe and North America [18]. Risk factors have included blood pressure, cigarette smoking, cholesterol (TC), LDL-C, HDL-C, and diabetes.

Access to essential noncommunicable disease medicines and basic health technologies in all primary health care facilities is essential to ensure that those in need receive treatment and counselling.

2.3.4 Study Motivation

Thanks to their machine learning-based core, chatbots and AI are naturally evolving day by day. The more qualified chatbots will be at understanding symptoms, the less often patients will need to make costly in-person visits to the doctor. Mean-

while, AI systems have the potential to make the healthcare industry more secure and efficient with the help of automation and data analysis.
Motivated by the growing interest on this evolving domain, in scope of the current study, we have implemented two different types of Chatbots using Dialogflow platform. We have also created an application on Heroku cloud platform, in order to establish a communication between our bot and the webhook fulfillments, as well as the Machine Learning model, that will be able to make predictions based on the data given from the end-user to the chatbot.

Chapter 3

Methodology

In this work, as already stated, the multifunctional role of chatbots in healthcare domain is examined in two different use cases, DrBot and Heart Risk chatbot. Both Chatbots have been implemented in Google's Dialogflow Platform and the backend application has been hosted in Heroku Cloud Platform.

3.1 Tools

3.1.1 Dialogflow

Dialogflow (formerly Api.ai, Speaktoit) is a Google-owned developer of human–computer interaction technologies based on natural language conversations. It is a natural language understanding platform used to design and integrate a conversational user interface into mobile apps, web applications, devices, bots, interactive voice response systems, and so on.
A Chatbot implemented in Dialogflow platform consists of the following core components:

Agents: A Dialogflow agent is a virtual agent that handles conversations with the end-users. It is a natural language understanding module that understands the nuances of human language. Dialogflow translates end-user text or audio during a conversation to structured data that apps and services can understand. This structured data is then passed to the relevant intents.

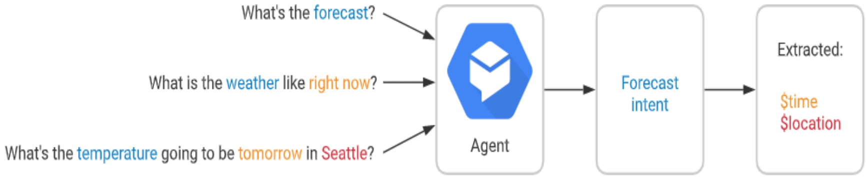

What's the forecast?
What is the weather like right now?
What's the temperature going to be tomorrow in Seattle?
Agent
Forecast intent
Extracted:
$time
$location

Figure 3.1: Agent and Intent matching

Intents: An intent categorizes an end-user's intention for one conversation turn. For each agent, many intents can be defined, where combined intents can handle a complete conversation. When an end-user writes or says something, referred to as an end-user expression, Dialogflow matches the end-user expression to the best intent in the agent. Matching an intent is also known as intent classification. Each intent can have follow-up intents, which decide how the conversation will continue. The default Dialogflow follow-up intents are: custom, fallback, yes, no, later, cancel, more, next, previous, repeat and select.number.

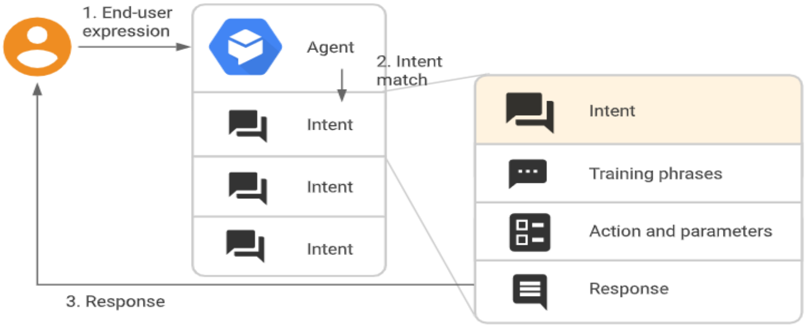


Figure 3.2: Dialogflow Intent matching

Entities: The term entity is used to describe the general concept of entities. Each intent parameter has a type, called the entity type, which dictates exactly how data from an end-user expression is extracted. The information received here is sent on for fulfilment.

Fulfillment: When fulfillment is enabled for an intent, Dialogflow responds to that intent by calling a service that the developer defines. When an intent with fulfillment enabled is matched, Dialogflow sends a request to the webhook service with information about the matched intent.

Contexts: Dialogflow contexts are similar to natural language context. For Dialogflow to handle ambiguous end-user expressions, it needs to be provided with context in order to correctly match an intent.

Contexts are divided to input and output and they are applied to intents. They work together to control the conversation flow: Output contexts control active contexts. When an intent is matched, any configured output contexts for that intent become active. Input contexts control intent matching. While contexts are active, Dialogflow is more likely to match intents that are configured with input contexts that are a subset of currently active contexts. With contexts, the user can : Control the order of intent matching, create context-specific intents with the same training phrases, control the flow of a conversation. Contexts can be configured for an intent by setting input and output contexts, which are identified by string names.[19]

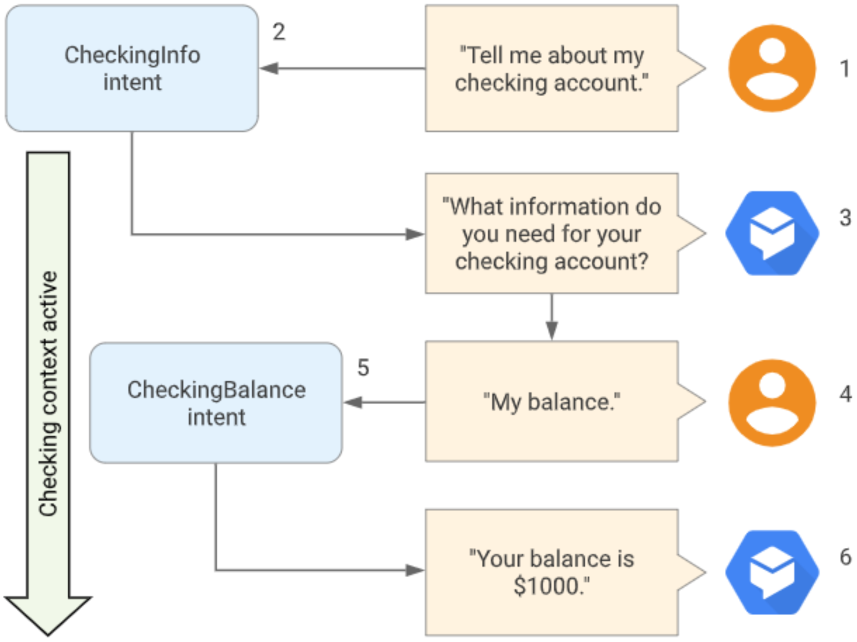


Figure 3.3: Dialogflow context matching

3.1.2 Infermedica API

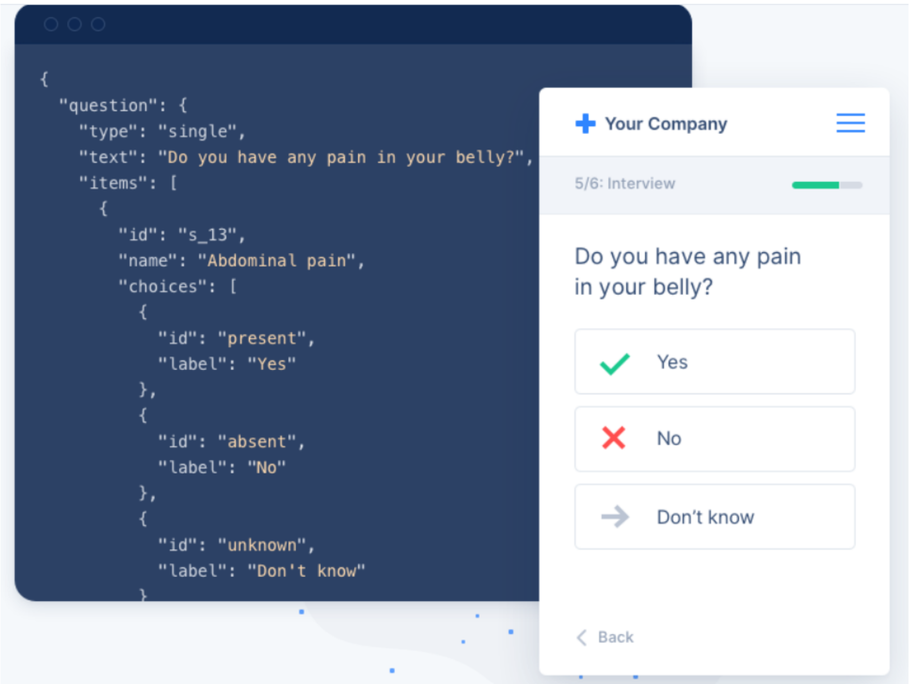
3.1.2.1 About Infermedica

Founded in 2012 in Poland, Infermedica is a health tech startup that offers an AI-driven platform for preliminary diagnosis and triage. Infermedica describes itself as an “AI-driven, customisable, multi-language” platform that aids patient care and healthcare service delivery.

In a nutshell, it provides an API for patient triage and preliminary medical diagnosis that can help developers implement an intelligent symptom checker or an adaptive patient intake form for any health application. If a patient’s health data (such as symptoms, risk factors or demographics) is sent to the API, its AI inference engine will analyze the data and provide the user with a list of likely conditions and relevant observations to verify. This is possible thanks to the sophisticated statistical algorithms used to perform diagnostic reasoning.

As it has been already mentioned, AI applications in Healthcare domain pose many challenges, mainly because of the lack of large public clean accurate datasets, due to privacy issues as well as the non-digitization of health till the recent years. Infermedica API has been our major enabler for creating a chatbot that will be able to perform accurate predictions, due to its sophisticated dataset background and algorithms.

More details about the API endpoints and its applications will be provided in the following sections, which will explain DrBot’s implementation.



```
{
  "question": {
    "type": "single",
    "text": "Do you have any pain in your belly?",
    "items": [
      {
        "id": "s_13",
        "name": "Abdominal pain",
        "choices": [
          {
            "id": "present",
            "label": "Yes"
          },
          {
            "id": "absent",
            "label": "No"
          },
          {
            "id": "unknown",
            "label": "Don't know"
          }
```

Figure 3.4: Infermedica API JSON and GUI representation

3.1.3 Heroku Cloud Platform

Heroku is a cloud platform that lets developers build, deliver, monitor and scale apps. It is a fast way to deploy apps in a URL, bypassing all complex infrastructure requirements. It is a secure platform which provides scalable database, web services and github integrations in order to run applications from the master repository of one's account. For the needs of this study, we integrated our github repos with Heroku, for both DrBot and Heart use case. Another important aspect of Heroku applications is that of providing extended logs in order to have an overview of the usage and bugs during runtime. In general, Heroku is an ecosystem that gives the opportunity to the end user to check the error logs of the runtime as well as to run live applications. This cloud platform system enabled us to host our backend code implementation.

3.1.4 Firebase Realtime Database

For the needs of the implementation of DrBot, a Realtime Database (RTDB) Application has been implemented. Google has been again our enabler by providing such an application through Firebase.

The Firebase Realtime Database is a cloud-hosted database. It lets the developer build rich, collaborative applications by allowing secure access to the database directly from client-side code. Data is persisted locally, and even while offline, realtime events continue to fire, giving the end user a responsive experience. When the device regains connection, the Realtime Database synchronizes the local data changes with the remote updates that occurred while the client was offline, merging any conflicts automatically.

The Realtime Database provides a flexible, expression-based rules language, called Firebase Realtime Database Security Rules, to define how data should be structured and when data can be read from or written to. When integrated with Firebase Authentication, developers can define who has access to what data, and how they can access it.

The Realtime Database is a NoSQL database and as such has different optimizations and functionality compared to a relational database [20].

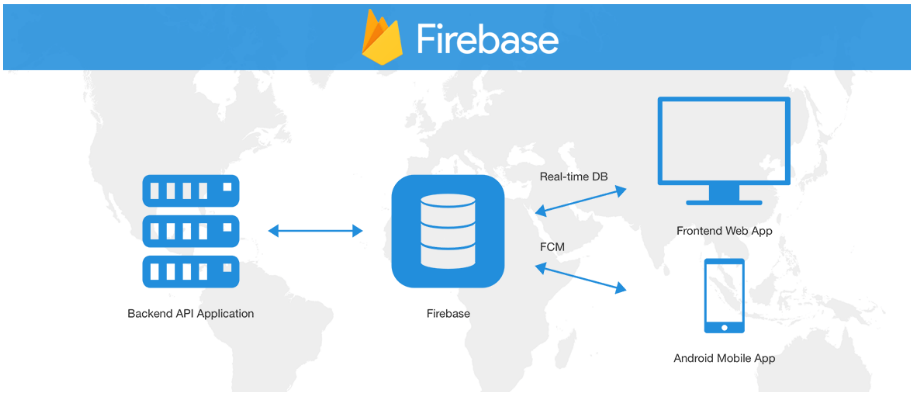


Figure 3.5: Firebase architecture

3.2 DrBot Use case

The first chatbot implemented as part of this study is DrBot.
DrBot is a medical symptom checker which asks for the user's age/sex and the symptoms he is experiencing. As a next step, an interview is conducted and according to his answers, DrBot analyzes the symptoms and provides him with clear, accurate information on potential causes, possible next steps and a triage suggestion. It can identify most issues that primary care doctors tend to see.

3.2.1 Architecture

The architecture of DrBot application consists of 3 main components: the Dialog Interface implemented in Dialogflow, the Python fulfillment application hosted in Heroku that integrates the bot with Infermedica API and the Firebase RTDB, which enables us to store the end user's answers and post requests to the API:

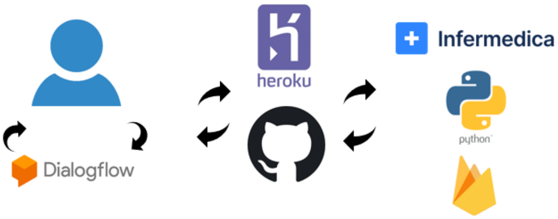


Figure 3.6: DrBot's Architecture

We will explain in detail in the following sections the implementation and functionality of each component separately, as well as their integration to the final DrBot application.

3.2.2 Infermedica API main endpoints

The most important part of DrBot's implementation is based on Infermedica API. Responsible for handling diagnostic reasoning, is the **diagnosis** endpoint, which accepts POST requests. Apart from sex and age, it requires a list of observed evidence (symptoms, risk factors or laboratory test results). The list cannot be

empty, so it first needs to gather some initial information about the case.
The response returned from diagnosis will contain three main sections:
the next diagnostic question to ask the user, a ranking of possible health conditions (**conditions** endpoint) and a flag indicating if the interview should be stopped (should_stop flag). The question attribute represents the next diagnostic question that can be presented to the user. Questions are follow-ups to already reported evidence and can be used to build conversation-like interfaces (a chatbot in our case) that resemble the way doctors interview their patients. There are three possible types of questions:

- Single: a yes/no question about a single symptom (e.g. “Do you have a headache?”)
- Group single: single-choice questions about a group of related symptoms (e.g. “What is your body temperature?”)
- Group multiple: a multiple-choice question about a group of related symptoms (e.g. “What is the character of your headache?”); any number of symptoms (including none) can be selected.

The conditions attribute is a list of health conditions related to the reported evidence. The list is a ranking sorted by the estimated probability of the conditions. DrBot accepts simple type questions, as it is recommended for chatbots or voice assistants where it is difficult or impossible to implement group questions.
It is possible and sometimes necessary (e.g. when analyzing data without user interaction) to settle for the condition ranking returned by the first call to diagnosis, but asking the user additional questions generated by the API can greatly improve the results.

Following on from the questions, the condition ranking seems to indicate that the patient may suffer from some condition, but answering additional questions could either help to confirm this suggestion, by increasing its probability and ruling out other, more serious health problems, or could prompt a broader differential diagnosis by suggesting other possible causes.[21]
The chatbot can continue this process: ask a question, accept an answer from the

user, append new evidence to the list, make a request to diagnosis. Each time the API will reply with the updated condition ranking and another question to ask.

Apart from the diagnosis endpoint, the Infermedica API provides a complementary **triage** endpoint that can categorize provided patient cases based on the seriousness of reported observations and the severity of likely conditions. This procedure may be viewed as similar to the telephone triage, hence the name of this endpoint.
The triage endpoint uses the same diagnostic engine that powers the diagnosis endpoint to compute the ranking of possible conditions. The triage classification algorithm proposed by Infermedica takes into account the severity of the most likely conditions identified by the diagnostic engine as well as the occurrence of any alarming symptoms or risk factors.
The default triage level is a five level triage. The values returned from the triage endpoint are the following, starting from the most severe:

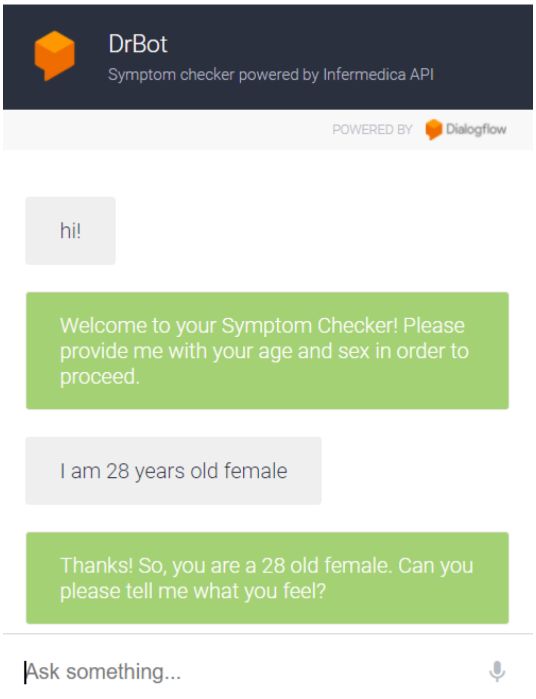
- *emergency_ambulance*: the reported symptoms are very serious and the patient may require emergency care. The patient should call an ambulance right now.
- *emergency*: the reported evidence appears serious and the patient should go to an emergency department. If the patient can't get to the nearest emergency department, he/she should call an ambulance.
- *consultation_24*: the patient should see a doctor within 24 hours. If the symptoms suddenly get worse, the patient should go to the nearest emergency department.
- *consultation*: the patient may require medical evaluation and may need to schedule an appointment with a doctor. If symptoms get worse, the patient should see a doctor immediately.
- *self_care*: a medical consultation is advised but not strictly required; the patient should observe his/her symptoms and consult a doctor if symptoms worsen within 24 hours.

Another endpoint that has been used for the implementation of DrBot is the **parse** endpoint, a custom Natural Language Processing technology, allowing applications

to understand clinical concepts (symptoms and risk factors) as mentioned by users as natural language text. The user's original message is sent to the endpoint to be processed and extract mentions of symptoms or risk factors. In our implementation, the user-described symptoms are sent through Dialogflow's fulfillment to parse and the symptoms are extracted.
All the above described information, as well as more details about the API and its functionalities can be found in the Infermedica API documentation [21].

3.2.3 Dialogflow

As we have already mentioned, a chatbot implementation in Dialogflow mainly consists of Agents, Intents and Entities. DrBot Agent consists of a Default Welcome intent, which as indicated by its name, welcomes the potential patient to the dialogue and is responsible for starting the conversation. The Default Welcome intents in general are triggered by the user, which is also the case for our agent :


DrBot

Symptom checker powered by Infermedica API

hi!

Welcome to your Symptom Checker! Please provide me with your age and sex in order to proceed.

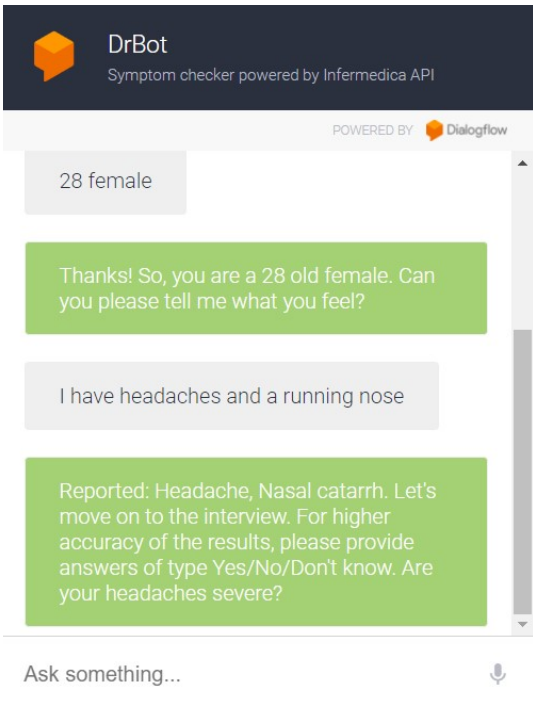
I am 28 years old female

Thanks! So, you are a 28 old female. Can you please tell me what you feel?

Figure 3.7: Default Welcome Intent

As it is shown above, we have replied to the Welcome intent's question so the next intent, which is the Age Sex intent will be triggered and the conversation flow will begin.

Since we have now collected the patient's sex and age, initial evidence must be collected as well (e.g. the patient's chief complaint and relevant risk factors). This action is part of the Answer Symptoms Intent, which waits for the user to provide the symptoms he is experiencing and uses this information to call the API's parse endpoint. Symptoms are extracted according to Infermedica's symptoms endpoint. In the below example, "running nose" is matched to "Nasal catarrh" symptom:



28 female

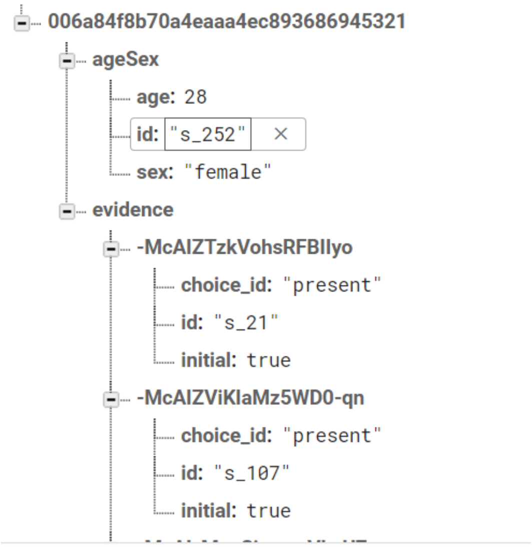
Thanks! So, you are a 28 old female. Can you please tell me what you feel?

I have headaches and a running nose

Reported: Headache, Nasal catarrh. Let's move on to the interview. For higher accuracy of the results, please provide answers of type Yes/No/Don't know. Are your headaches severe?

Figure 3.8: Symptoms Intent

We have now reached a part in the conversation where the inital evidence has been gathered from the potential patient and the interview flow with the followup questions can start. Responsible for this action is the Answer Follow Up Question Intent. Through our Python fulfillment of this intent, this initial information stored in our RTDB is sent as a request to the diagnosis endpoint. Each time a user sends a reply to the API's question, the JSON object is updated with the relevant information regarding the presence or not of a symptom and if it has been initially reported or not. An example is presented in the figure below:

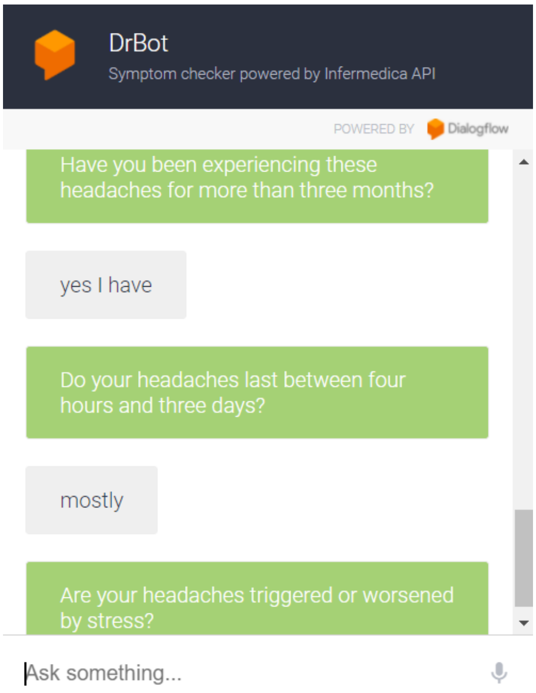


```
006a84f8b70a4eaaa4ec893686945321
    ageSex
        age: 28
        id: "s_252"
        sex: "female"
    evidence
        -McAIZTzkVohsRFBIlyo
            choice_id: "present"
            id: "s_21"
            initial: true
        -McAIZViKlaMz5WD0-qn
            choice_id: "present"
            id: "s_107"
            initial: true
```

Figure 3.9: RTDB JSON record

The response to this first request will contain a diagnostic question that should be presented to the patient. The patient's answer should then be added to the list of already collected evidence (RTDB update). The process should continue in the following manner:

- send a request to diagnosis with the updated evidence list.
- ask the patient the question returned from diagnosis.
- add the patient's answer to the existing evidence list.
- repeat the steps.



DrBot

Symptom checker powered by Infermedica API

Have you been experiencing these headaches for more than three months?

yes I have

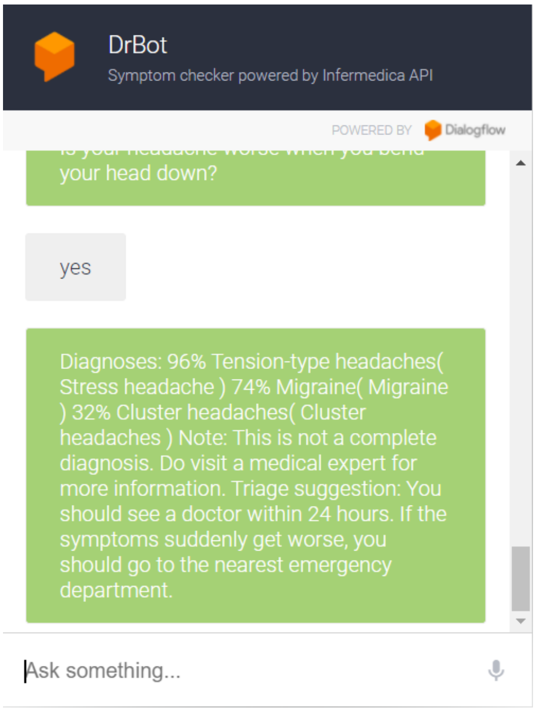
Do your headaches last between four hours and three days?

mostly

Are your headaches triggered or worsened by stress?

Figure 3.10: Follow Up Questions

The process can continue for as long as necessary, until a stop condition is met. The should_stop attribute offers a convenient stop recommendation that should be used in most cases, according to Infermedica's documentation. In general, the number of questions answered and the probability of the top conditions in the ranking should be considered when deciding when to stop the interview. By the time this condition is met a diagnosis is returned to the end user, with probabilities of possible conditions along with the conditions' common names, as well as a triage suggestion. An example can be seen below:



DrBot

Symptom checker powered by Infermedica API

Is your headache worse when you bend your head down?

yes

Diagnoses: 96% Tension-type headaches(Stress headache) 74% Migraine(Migraine) 32% Cluster headaches(Cluster headaches) Note: This is not a complete diagnosis. Do visit a medical expert for more information. Triage suggestion: You should see a doctor within 24 hours. If the symptoms suddenly get worse, you should go to the nearest emergency department.

Figure 3.11: Diagnosis and Triage Suggestion

3.2.4 Fulfillment

As already stated, DrBot is an application that requires the integration of different systems and applications. Dialogflow provides the developer with the Fulfillment component, enabling the aforementioned integration. By default, the agent responds to a matched intent with a static response. If one of the integration options is used, a more dynamic response can be provided by using fulfillment. When fulfillment is enabled for an intent, Dialogflow responds to that intent by calling a service that the developer defines. This service is a webhook service in our case. When an intent with fulfillment enabled is matched, Dialogflow sends a request to the webhook service with information about the matched intent. The system can perform any required actions and respond to Dialogflow with information on how to proceed. The following diagram borrowed from Dialogflow Documentation [19] shows the processing flow for fulfillment:

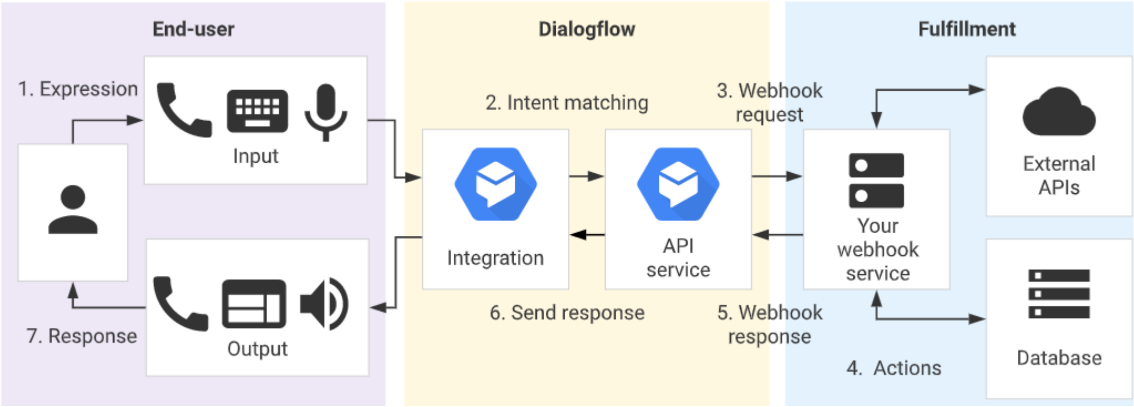


Figure 3.12: Dialogflow Fulfillment example

DrBot's fullfillment service is deployed in Heroku Cloud Platform. The github repository that contains the relevant implementation is integrated with Heroku, so as to host our backend service implementation. The code has been implemented in Python programming language by using the guidelines provided by Infermedica for building a symptom checker.

3.3 Coronary Heart Disease Risk Use case

The second conversational agent that was implemented as part of this study was the CHD Risk agent. This agent collects information regarding the user's demographic, behavioural and medical risk factors, communicates the information with a real-time machine learning algorithm and returns a prediction result.

3.3.1 Dataset

The dataset that was used to train our Machine Learning model for the CHD Risk use case is publicly available on Kaggle website, and it is from an ongoing cardiovascular study on residents of the town of Framingham, Massachusetts [22]. The classification goal is to predict whether the patient has 10-year risk of future coronary heart disease (CHD). The dataset includes over 4,000 records and 15 attributes, each of which is a potential risk factor. There are demographic, behavioral as well as medical risk factors:

Demographic:

- Sex: male or female(Nominal)
- Age: Age of the patient;(Continuous - Although the recorded ages have been truncated to whole numbers, the concept of age is continuous)
- Education: Patient's education level

Behavioral:

- Current Smoker: whether or not the patient is a current smoker (Nominal)
- Cigs Per Day: the number of cigarettes that the person smoked on average in one day(Continuous)

Information on medical history:

- BP Meds: whether or not the patient was on blood pressure medication (Nominal)
- Prevalent Stroke: whether or not the patient had previously had a stroke (Nominal)
- Prevalent Hyp: whether or not the patient was hypertensive (Nominal)
- Diabetes: whether or not the patient had diabetes (Nominal)

Information on current medical condition:

- Tot Chol: total cholesterol level (Continuous)
- Sys BP: systolic blood pressure (Continuous)
- Dia BP: diastolic blood pressure (Continuous)
- BMI: Body Mass Index (Continuous)
- Heart Rate: heart rate (Continuous - In medical research, variables such as heart rate though in fact discrete, yet are considered continuous because of large number of possible values.)
- Glucose: glucose level (Continuous)

3.3.2 Preparation

3.3.2.1 General

In order to prepare the dataset for our machine learning algorithms, we performed an exploratory analysis. Our fist step was to check for null or missing values, as well as duplicate ones. The percentage of missing data was only 12.74%, so we have dropped those rows. No duplicate values appeared on the dataset. Then, according to the correlation matrix we dropped the education feature, as it appears to have no correlation at all with the risk of 10 year CHD.

3.3.2.2 Feature extraction and scaling

Our next step was the feature extraction, in order to select which are the ones with the highest importance for the outcome variable of CHD risk. The objective of feature selection is to select a smaller group of the initial features (15 in our case) and still achieve similar risk prediction accuracy, in order to achieve more interpretable models and faster training.

We applied scikit learn's SelectKBest feature extraction, with an ANOVA F-value score function in order to get the 10 most prominent features. SelectKBest gives the best k features based on the ratio between variances values. Importantly, ANOVA is used when one variable is numeric and one is categorical, such as numerical input variables and a classification target variable in a classification task. The features that resulted as the most prominent are the following:
sysBP, glucose, age, totChol, cigsPerDay, diaBP, prevalentHyp, diabetes, BP Meds, male.
Feature scaling in machine learning is one of the most critical steps during the preprocessing of data before training a model. The most common techniques of feature scaling are Normalization and Standardization.
Normalization is a technique often applied as part of data preparation for machine learning. The goal of normalization is to change the values of numeric columns in

the dataset to use a common scale, without distorting differences in the ranges of values or losing information. Normalization is also required for some algorithms to model the data correctly.

The result of standardization (or Z-score normalization) is that the features will be rescaled to ensure the mean and the standard deviation to be 0 and 1, respectively. In our implementation we performed a MinMax Scaling Normalization. MinMaxScaler preserves the shape of the original distribution, doesn't reduce the importance of outliers and the range of the feature returned is from 0 to 1.

3.3.2.3 Class imbalance

Examining the values of the target variable of 10 year CHD risk, we observed that there is a high class imbalance, as 3179 instances did not suffer from CHD, while only 572 did. The challenge of working with imbalanced datasets is that most machine learning techniques will ignore, and in turn have poor performance on, the minority class, although typically it is performance on the minority class that is most important. In general, one way to solve this problem is to oversample the examples in the minority class.

This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance the class distribution but does not provide any additional information to the model.

In order to address the above problem, we implemented the Synthetic Minority Oversampling Technique (SMOTE). This technique was described by Nitesh Chawla, et al. in their 2002 paper named for the technique titled "SMOTE: Synthetic Minority Over-sampling Technique." [23]. The original paper on SMOTE suggested combining SMOTE with random undersampling of the majority class.

The imbalanced-learn Python library supports random undersampling via the RandomUnderSampler class. We have updated the dataset to first oversample the minority class to have 80 percent the number of examples of the majority class ,then used random undersampling to reduce the number of examples in the majority class. We then chained these two transforms together into a Pipeline. The Pipeline can then be applied to a dataset, performing each transformation in turn and returning a final dataset with the accumulation of the transform applied to it, in this case oversampling followed by undersampling.

The below figure shows the classes before and after balancing:

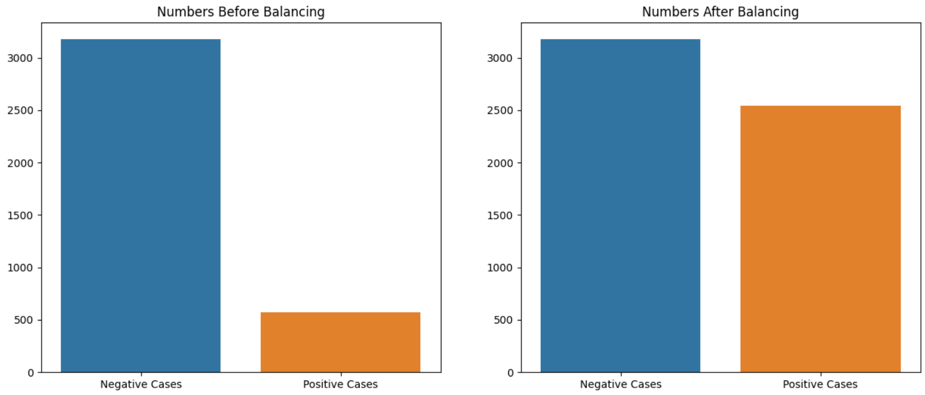


Figure 3.13: Classes before and after SMOTE

3.3.3 Machine Learning

In machine learning, classification refers to a predictive modeling problem where a class label is predicted for a given example of input data. It is a supervised learning approach (i.e. part of the data should have labels) at which a trained model is used to make predictions concerning a new, unseen to the model dataset. An observation can be classified into two (binary) or more classes (multi-class). In this study we aim for binary classification, as we would like to classify the probability for a patient of having or not a 10 year CHD risk. In order to get the aforementioned prediction, we tested the traditional machine learning models to figure out which one may better fit to our problem. We tested the following algorithms:

- Logistic Regression
- Decision Trees
- K Nearest Neighbors
- Support Vector Machines

Logistic Regression:

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables [24]. It is much similar to the Linear Regression except the way they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving classification problems.

Decision trees:

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation. A decision node can have one or more branches whereas a leaf node is the node that represents a classification label or a decision. The root node of the tree is the node that corresponds to the best predictor.

K Nearest Neighbors:

The principle behind nearest neighbor methods is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. The number of samples can be a user-defined constant (k-nearest neighbor learning), or vary based on the local density of points (radius-based neighbor learning) [25]. The distance can, in general, be any metric measure: standard Euclidean distance is the most common choice. Neighbors-based methods are known as non-generalizing machine learning methods, since they simply "remember" all of its training data.

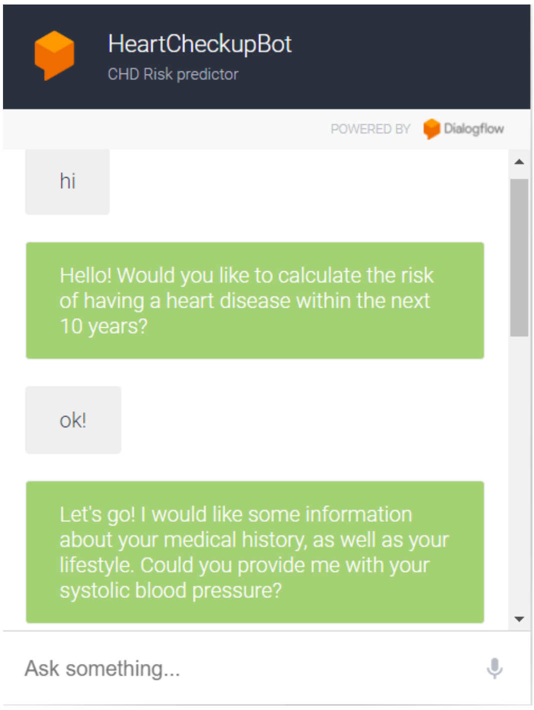
Support Vector Machines:

Support vector machines (SVM) method is a supervised learning algorithm that is used for solving classification or regression problems, that maybe linear or nonlinear. Initial work for its development had been done by Vapnik-Lerner, 1963, with the introduction of the Generalized Portrait algorithm and further development was done by Vapnik-Chervonenkis, 1964, regarding the same algorithm, that constructs separating hyperplanes with maximum margin. The objective of this algorithm is to find a hyperplane in an N-dimensional space, where N is the number of features that distinctly classify the data points.

For our experiments with all the above classifiers we have used the Scikit-learn Python library.

3.3.4 Dialogflow

Heart Checkup Agent consists of a Default Welcome Intent, which as indicated by its name, welcomes the user to the dialogue and is responsible for starting the conversation. The Default Welcome intents in general are triggered by the user, which is also the case for our Heart agent:



HeartCheckupBot

CHD Risk predictor

hi

Hello! Would you like to calculate the risk of having a heart disease within the next 10 years?

ok!

Let's go! I would like some information about your medical history, as well as your lifestyle. Could you provide me with your systolic blood pressure?

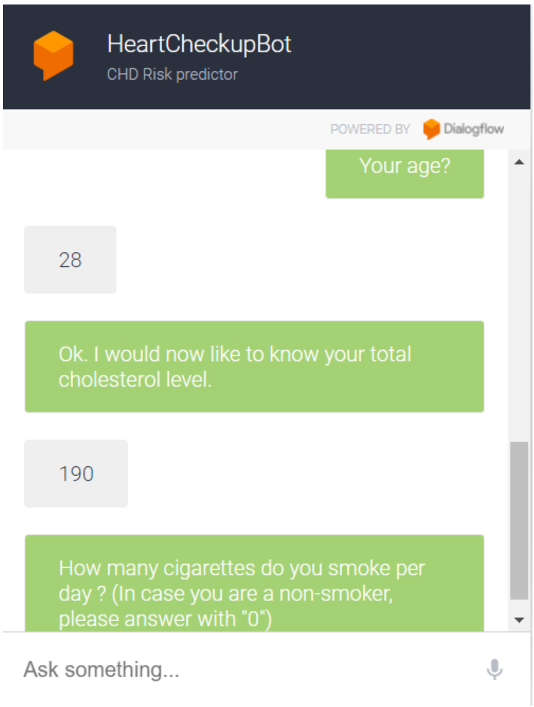
Figure 3.14: Default Welcome Intent - Heart risk predictor

As it is shown above, we have given a positive reply to the Welcome intent's question so the next intent, which is the Heart Intent will be triggered and the conversation flow will begin.

The Heart Intent asks for each parameter that should be passed as feature to the machine learning algorithm. Those features are the following:

- Systolic Blood Pressure
- Glycose
- Age
- Total Blood cholesterol
- Number of Cigarettes smoked per day
- Diastolic Blood Pressure
- If the user is hypertensive
- If the user has diabetes
- If the user is on blood pressure medication
- Sex

The intent will ask for each feature value, in order to collect them all and provide them to the machine learning model. Below an example of the conversation flow:



HeartCheckupBot

CHD Risk predictor

Your age?

28

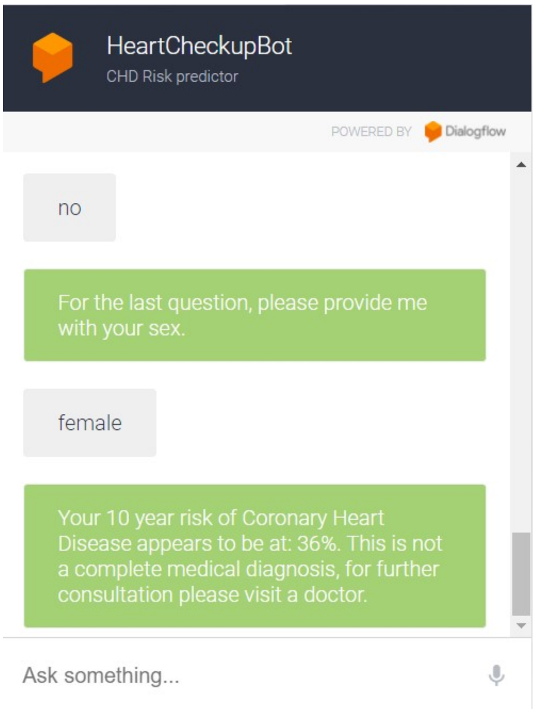
Ok. I would now like to know your total cholesterol level.

190

How many cigarettes do you smoke per day ? (In case you are a non-smoker, please answer with "0")

Figure 3.15: Heart Intent - Parameter values collected

By the time all the parameters are gathered by our bot, they are sent to our webhook fulfillment and provided as input to the already trained and saved machine learning model. The prediction is made and the relevant risk is prompted to the end-user, as we can see below:



HeartCheckupBot
CHD Risk predictor

no

For the last question, please provide me with your sex.

female

Your 10 year risk of Coronary Heart Disease appears to be at: 36%. This is not a complete medical diagnosis, for further consultation please visit a doctor.

Figure 3.16: Heart Intent - Heart risk prediction

3.3.5 Fulfillment

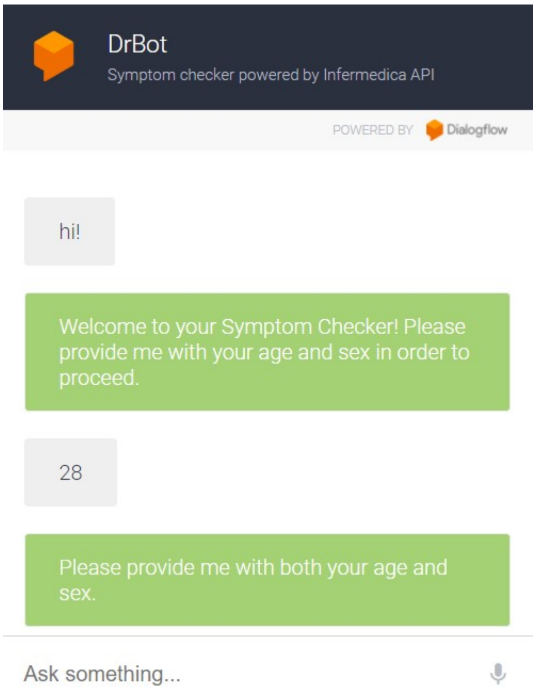
The same methodology as DrBot's fulfillment has been followed for Heart Risk bot as well. The webhook service is deployed in Heroku Cloud Platform. The github repository that contains the relevant implementation is integrated with Heroku, so as to host our backend service implementation. The webhook method receives the parameters from Dialogflow and passes them to the ML model, which is already trained and saved as pickle file. The code has been implemented in Python Programming Language.

Chapter 4

Experiments and Evaluation

4.1 DrBot

In order to evaluate DrBot's usability and results, we have tested it with the support of 12 volunteer users, aged between 28 and 66. The results were satisfying, as none of them had problems communicating with the bot and conducting a normal conversation. In cases of miss-typing or invalid data given (e.g. age $<$ 14 or age $>$ 100 which is the threshold we have set) the bot was able to re-prompt the user to re-enter an answer, following the relevant instructions. Below an example where the user has provided only his age without sex, and an example where he has provided an invalid value for age:



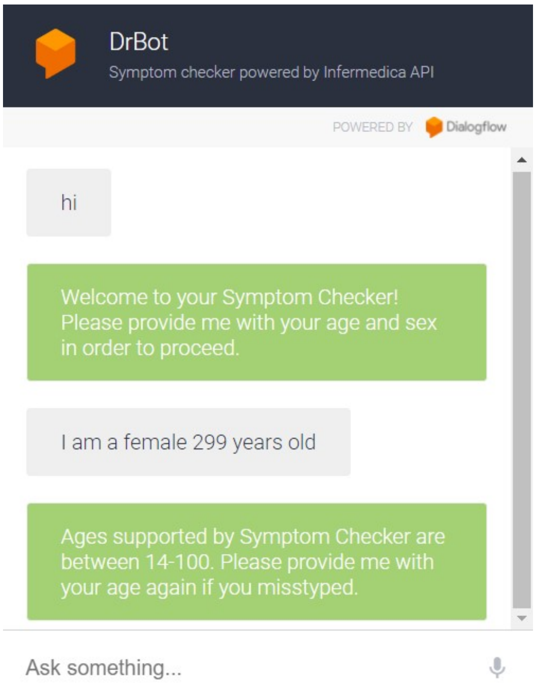
hi!

Welcome to your Symptom Checker! Please provide me with your age and sex in order to proceed.

28

Please provide me with both your age and sex.

Figure 4.1: Age Sex Intent - Sex not provided



hi

Welcome to your Symptom Checker! Please provide me with your age and sex in order to proceed.

I am a female 299 years old

Ages supported by Symptom Checker are between 14-100. Please provide me with your age again if you misstyped.

Figure 4.2: Age Sex Intent - Invalid age provided

All of the users that tested the bot were able to reach the end of the conversation and receive a diagnosis and triage suggestion.

We have tested many cases providing various symptoms to the chatbot and giving ambiguous answers to the follow up questions. We have observed that the more we were trying to "confuse" the chatbot, the more it was prompting extra questions for clarifying the relevant case, which is a satisfying result.
What we have noticed as a potential point of improvement, is that at some point DrBot asks disjunctive questions like "Is your headache stabbing **or** throbbing?". Since we have used the single group type questions, which is suggested by Infermedica when implementing a conversational agent, the bot accepts answers only of type Yes/No/Unknown. However, questions containing the word "or" have confused 5 out of 12 users to chose one of the 2 options and answer "stabbing", for example to the above question. Even on that case, the bot re-prompted the user to answer with the expected types and the conversation continued without ending unexpectedly. This issue could have been addressed with an implementation of a richer User Interface (UI), prompting the user with fixed answers to chose.

4.2 Heart Risk Chatbot

4.2.1 Experiments

The experiments and evaluation of Heart Risk bot mostly concerned the ML algorithms. As already mentioned, we performed experiments on 4 different ML algorithms in order to fine-tune their hyper-parameters and decide which one had the best results and should be saved. The algorithms tested were: Logistic Regression, Decision Tree Classifier, K Nearest Neighbours and Support Vector Machines.

A model hyper-parameter is a characteristic that is external to the model and whose value cannot be estimated from data. The value of the hyper-parameter has to be set before the learning process begins. Grid-search is used to find the optimal hyperparameters of a model which results in the most ‘accurate’ predictions. In our study we performed Grid-Search, using the relevant scikit-learn Python library.

4.2.2 Evaluation

To select the most efficient algorithm that would be deployed to the cloud platform we experimented on multiple trainings and evaluations. The metrics that were examined were accuracy, precision, recall, f1 score and the ROC AUC Curve.

Accuracy

Accuracy is one metric for evaluating classification models. Informally, accuracy is the fraction of predictions our model got right. Formally, accuracy has the following definition:

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Number of total predictions}}$$

For binary classification, accuracy can also be calculated in terms of positives and negatives as follows:

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

where TP, TN, FP, FN correspond to True Positives, True Negatives, False Positives and False Negatives accordingly.

Accuracy is not always the right evaluation metric, especially when working with a class-imbalanced data set, where there is a significant disparity between the number of positive and negative labels.

Precision and Recall

Precision attempts to answer the following question: What proportion of positive identifications was actually correct?

Precision is defined as follows:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall attempts to answer the following question: What proportion of actual positives was identified correctly?

Mathematically, recall is defined as follows:

$$Recall = \frac{\text{TP}}{\text{TP + FN}}$$

F1-Score

In statistical analysis of binary classification, the F-score or F-measure is a measure of a test's accuracy. It is calculated from the precision and recall of the test. The F1 score is the harmonic mean of the precision and recall:

$$F1 - Score = 2 * \frac{\text{precision * recall}}{\text{precision + recall}}$$

ROC AUC Curve

When making a prediction for a binary or two-class classification problem, there are two types of errors that we could make: False Positives and False Negatives. A common way to compare models that predict probabilities for two-class problems is to use a ROC curve. A ROC Curve (Receiver Operating Characteristic Curve) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters: True Positive Rate (also known as recall) and False Positive Rate (FPR). FPR is defined as follows:

$$FPR = \frac{\text{FP}}{\text{FP + TN}}$$

A ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives.

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve from (0,0) to (1,1).

The scores of each algorithm, as well as the ROC AUC Curves are presented below:

Table 4.1: Evaluation Metrics of Classification Models

Metrics	Logistic Regression	Decision Trees	KNN	SVM
Accuracy	0.61	0.74	0.87	0.91
Precision	0.67	0.72	0.79	0.88
Recall	0.60	0.67	0.93	0.92
F1 Score	0.57	0.69	0.87	0.90

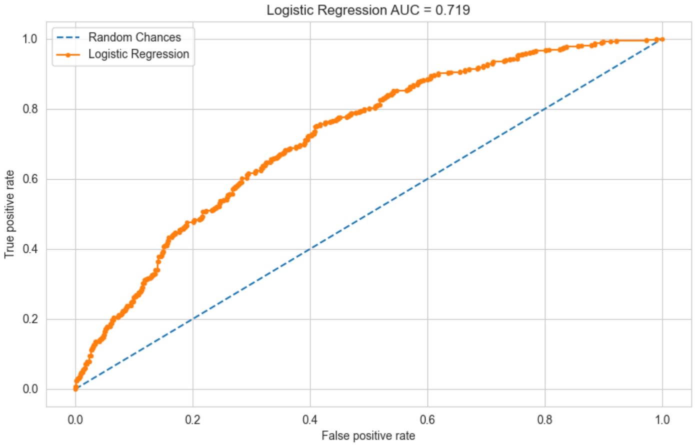


Figure 4.3: Logistic Regression ROC Curve

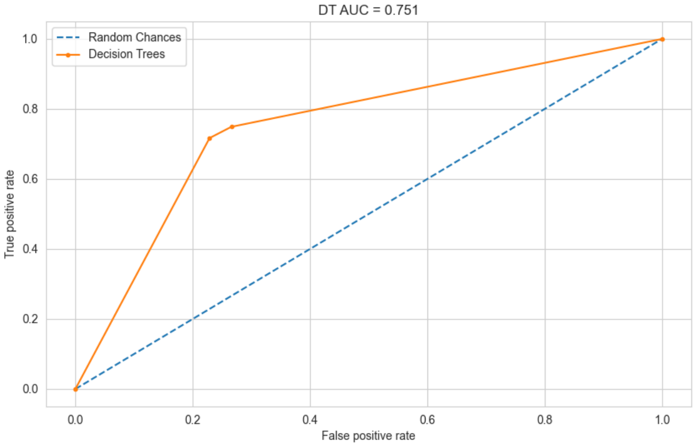


Figure 4.4: Decision Trees ROC Curve

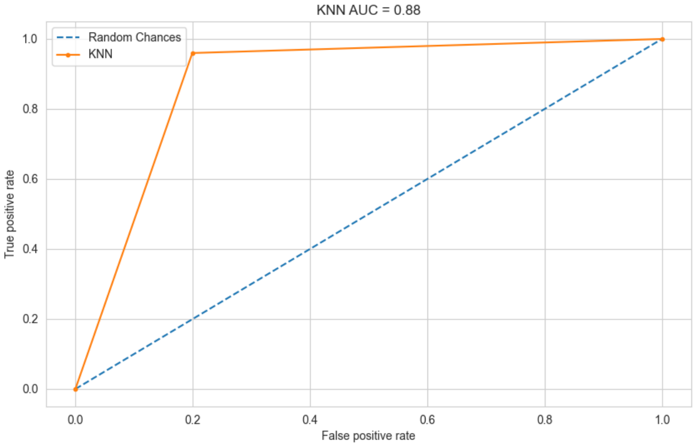


Figure 4.5: KNN ROC Curve

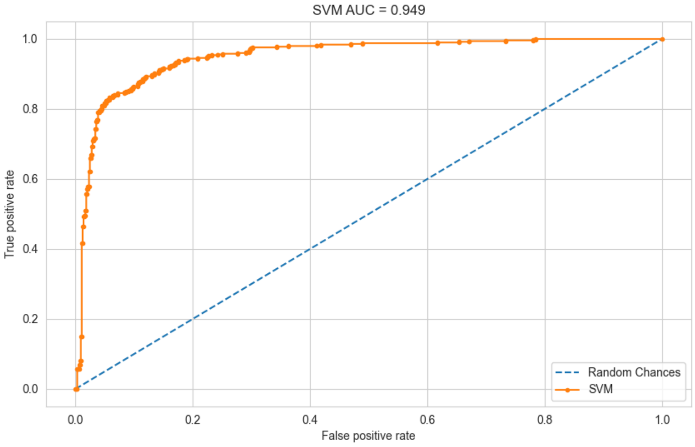


Figure 4.6: SVM ROC Curve

Since our case concerns health predictions, it was considered very critical to minimize both false negatives and false positive predictions. Taking into consideration all the evaluation metrics as well as the ROC Curve plots, we came to the conclusion that SVM algorithm as the most robust in our case study.

Chapter 5

Discussion

In this master thesis, we have examined the multifunctional role of chatbots in two different use cases in Healthcare domain.

DrBot is our first chatbot implemented in Google Dialogflow Platform. Its purpose is to provide AI-enable general diagnosis and triage suggestions, powered by Infermedica's API and AI inference models.

DrBot has been tested by 12 volunteers, leading to very satisfying results. All of the users were able to conduct a full conversation and receive a diagnosis and triage suggestion, according to the original symptoms reported as well as their answers through the interview flow. Interesting appears to be the fact that even though sometimes the conversation lasted longer than the users' expectations, they were not discouraged to quit the process and the interview kept their interest going. This is a very encouraging observation for the future development of chatbots in Healthcare domain, since conversational agents in general seem to be adopted more and more by organizations and companies in other domains, while in Healthcare some concerns are still raised.

HeartRisk chatbot is our second use case, also implemented in Google Dialogflow Platform. We have implemented our own machine learning model for this bot, which receives the parameters through Dialogflow's webhook requests and calculates the potential risk of the user of suffering from CHD in 10 years. Four classification algorithms have been fine-tuned and tested: Logistic Regression, Decision Trees, K-Nearest Neighbours and Support Vector Machines. The most robust and accurate according to a combination of evaluation metrics was SVM, which was the one that we saved as our final model.

What appears to be challenging in such cases, where a machine is called to predict such a serious risk as that of suffering from an illness, is that the results must achieve the highest accuracy possible. Especially when the false negative cases are concerned, we have to reassure the user that our model's predictions are the most accurate. We wouldn't like our agent to comfort a potential patient that he is of a low risk, when he actually has a high potential of suffering from an illness. Developers of AI algorithms must be vigilant to potential dangers, including dataset shift, unintended bias, the challenges of generalisation to new populations and the unintended negative consequences of new algorithms on health outcomes [26].

Chapter 6

Conclusions and Future Work

The aim of this master thesis was to study the current applications of chatbots in Healthcare domain and examine two different implementations: AI-enabled general diagnosis and medical support in scope of a potential disease risk prediction. DrBot, responsible for conducting an interview and providing potential diagnosis and triage suggestion is an example of a conversational agent that could be possibly adopted by hospitals or other healthcare organizations. During challenging times and circumstances, like the one we are traversing since COVID-19 outbreak, a conversational agent can be both time and resource-saving as a significant number of potential patients can be attended on directly, without having to schedule an appointment with a doctor or any clinical institution. Future work regarding this area could involve the chatbot implementation in different languages in order to cover a larger geographical space, as well as an implementation considering empathic dimensions in the conversation.

Heart Risk chatbot collects some target medical and lifestyle information from users, aiming to perform a prediction regarding a heart disease risk. After implementing and testing four different classification algorithms, we achieved a high prediction accuracy score with SVM algorithm. This high accuracy obtained allowed us to conclude that similar ML/AI methods seem to be an alternative to more expensive laboratory tests, at least during the initial moment of the CHD risk assessment. In conjunction with a conversational agent providing the questionnaire in the form of a chat, an initial heart check can be provided easily and more user-friendly, avoiding the laboratory examination at least in the first place. Future work could be focused on training the ML algorithms with a more diverse dataset, in terms of the variety of age and life habit profiles of the volunteers, taking also into consideration the fact that the dataset was highly imbalanced. Moreover, future work can include training neural networks and performing Deep Learning techniques on larger datasets, in order to achieve the highest accuracy possible.

References

[1] Chandra Khatri, Behnam Hedayatnia, Anu Venkatesh, Jeff Nunn, Yi Pan, Qing Liu, Han Song, Anna Gottardi, Sanjeev Kwatra, Sanju Pancholi, Ming Cheng, Qinglang Chen, Lauren Stubel, Karthik Gopalakrishnan, Kate Bland, Raefer Gabriel, Arindam Mandal, Dilek Hakkani-Tur, Gene Hwang, and Rohit Prasad. Advancing the state of the art in open domain dialog systems through the alexa prize, 12 2018.

[2] Michael McTear. 2020.

[3] The Medical Futurist. The top 12 health chatbots, 2020.

[4] Alaa Abd-alrazaq, Mohannad Alajlani, Ali Alalwan, Bridgette Bewick, Peter Gardner, and Mowafa Househ. An overview of the features of chatbots in mental health: A scoping review. *International Journal of Medical Informatics*, 132:103978, 09 2019.

[5] Juanan Pereira and Oscar Díaz. Using health chatbots for behavior change: A mapping study. *Journal of Medical Systems*, 43, 04 2019.

[6] Lorainne Tudor Car, Dhakshenya Dhinagaran, Bhone Kyaw, Tobias Kowatsch, Shafiq Joty, Yin-Leng Theng, and Rifat Atun. Conversational agents in health care: Scoping review and conceptual analysis. *Journal of Medical Internet Research*, 22:e17158, 08 2020.

[7] Tom Nadarzynski, Oliver Miles, Aimee Cowie, and Damien Ridge. Acceptability of artificial intelligence (ai)-led chatbot services in healthcare: A mixed-methods study. *DIGITAL HEALTH*, 5:205520761987180, 08 2019.

[8] F. Amato, S. Marrone, V. Moscato, Gabriele Piantadosi, A. Picariello, and C. Sansone. Chatbots meet ehealth: Automatizing healthcare. In *WA-IAH@AI*IA*, 2017.

[9] Clark C Robinson KJ. Beaudry J, Consigli A. Getting ready for adult healthcare: Designing a chatbot to coach adolescents with special health needs through the transitions of care. In *J Pediatr Nurs.*, 2019.

[10] Fakih Habib, Ghare Shakil, Shaikh Iqbal, and Shaikh Sajid. *Self-Diagnosis Medical Chatbot Using Artificial Intelligence*, pages 587–593. 01 2021.

[11] Antoine Piau, Rachel Crissey, Delphine Brechemier, Laurent Balardy, and Fati Nourhashemi. A smartphone chatbot application to optimize monitoring of older patients with cancer. *International Journal of Medical Informatics*, 128, 05 2019.

[12] Chin-Yuan Huang, Ming-Chin Yang, Chin-Yu Huang, Yu-Jui Chen, Meng-Lin Wu, and Kai-Wen Chen. A chatbot-supported smart wireless interactive healthcare system for weight control and health promotion. pages 1791–1795, 12 2018.

[13] Keerthan Kumar T G. The companion chatbot for dementia patients. *International Journal of Advanced Science and Technology*, 29:6582–6592, 07 2020.

[14] Marzyeh Ghassemi, Tristan Naumann, Peter Schulam, Andrew Beam, Irene Chen, and Rajesh Ranganath. A review of challenges and opportunities in machine learning for health. *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science*, 2020:191–200, 05 2020.

[15] Wikipedia. Medical diagnosis, 2021.

[16] Adam Zagorecki, Piotr Orzechowski, and Katarzyna Hołownia. A system for automated general medical diagnosis using bayesian networks. *Studies in health technology and informatics*, 192:461–5, 08 2013.

[17] World Health Organization. Cardiovascular diseases.

[18] Paul G McGovern, James S Pankow, Eyal Shahar, Katherine M Doliszny, Aaron R Folsom, Henry Blackburn, and Russell V Luepker. Recent trends

in acute coronary heart disease—mortality, morbidity, medical care, and risk factors. *New England Journal of Medicine*, 334(14):884–890, 1996.

[19] Google. Dialogflow ES Documentation.

[20] Google. Firebase Documentation.

[21] Infermedica. Infermedica API Documentation.

[22] NIH National Heart Institute. Framingham Heart Study-Cohort.

[23] Nitesh Chawla, Kevin Bowyer, Lawrence Hall, and W. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Intell. Res. (JAIR)*, 16:321–357, 06 2002.

[24] Marlene Müller. Generalized linear models. 02 2004.

[25] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[26] Christopher Kelly, Alan Karthikesalingam, Mustafa Suleyman, Greg Corrado, and Dominic King. Key challenges for delivering clinical impact with artificial intelligence. *BMC Medicine*, 17, 12 2019.