Intelligent Assisted Living in Pregnancy

by

S. M. Tanveer Zaman
18101539
Rafia Tasneem
18101288
Tasmia Shakerin
18101663

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
January 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

S. M. Tanveer Zaman
18101539

Rafia Tasneem
18101288

Tasmia Shakerin
18101663

Approval

The thesis/project titled "Intelligent Assisted Living in Pregnancy" submitted by

1. S. M. Tanveer Zaman(18101539)
2. Rafia Tasneem(18101288)
3. Tasmia Shakerin(18101663)

Of Fall, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on January 20, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Pregnancy period has always been the most sensitive and tenacious phase for the mothers to go through. Despite living in this modern world, it is still difficult to fight with the distress and complications of pregnancy, and that the rate of unfortunate miscarriages and stunting are still seen to be rising in this era. The huge advancement of technology, however, has stood up as a shield to protect and prevent these misfortunes from taking place, at a very early stage, by taking some of its useful measures. Artificial Intelligence is playing a vital role in this case and is able to serve aid in this regard. With the help of these techniques, we have decided and attempted to come up with a proposal of a system that is smart enough to assist like a doctor. Taking account of all the minor and major symptoms and difficulties that a woman faces during her pregnancy period, our system will provide an effective outcome based on the information provided by that mother and will encourage the person to take necessary actions. It is often seen that the health issues that a woman faces during her pregnancy period is considered lightly since a visit to the doctor can prove costly. Considering these issues, our proposed application will work its best to instantly impart the solutions for all the common difficulties that a mother may go through during this tough time. This smart proposed application, therefore, aims to conduct operations based on diagnosing the complications the pregnant women face with the data of the symptoms they are showing while also predicting possible miscarriage or severe health complications.

Keywords: Data Mining; Machine Learning; Pregnancy; Miscarriage; Complications; Prediction; Decision tree; Linear Regression Analysis; Chatbot

Dedication (Optional)

Our research is dedicated to all the mothers out there who happily accept such difficulties and hardships in order to not only give safe birth to their offspring but also take care of them with unparalleled love and care throughout their lives. No life in this world would have been able to make it here without the willing sacrifices that they make out of their unconditional love for their child.

We would also love to dedicate this paper especially to our mothers who have always put our needs and wants above their own and dedicated all their time and affection to us ever since they conceived us.

Acknowledgement

Firstly, all praise to the Almighty Allah for whom our thesis have been completed without any major interruption. Secondly, to our supervisor Dr. Golam Rabiul Alam Sir for his kind support and advice in our work. He helped us whenever we needed help. We are truly grateful to Sir for his unwavering faith and support on us which guided us in learning new things and overcoming the challenges we faced during our research period. Thirdly, we want to sincerely thank our respected faculty members for their aid and encouragement throughout our stay at BRAC University. This has truly been a blessing for us. Finally to our parents, without whose throughout support it may not be possible. With their kind support and prayer we are now on the verge of our graduation.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AI Artificial Intelligence

FFN Feed Forward Network

ML Machine Learning

MSF Mother's Significant Feature (MSF)

ROC Receiver Operating Characteristic

Chapter 1

Introduction

1.1 Background

Medical service has always been one of the major needs in our everyday life since accidents, emergencies and illness may crop up anytime and urgent treatment might be needed to save one's life. With the rise of population in a society, the need and importance of healthcare service is rising accordingly and when considering the birth of a new child, medical facilities turn out to be the most important and serious issues that can never be avoided in this regard.

Pregnancy period has always been the most sensitive phase, a mother has to go through with profound physiological changes. In this tough time, mothers need to be guided and treated properly and all the necessary measures including their diet, nutrition, exercise, diagnosis records, nearby clinic, doctor's profession and availability must be carefully noted and maintained as a caution in keeping clear of difficulties. It is often seen that women still lack consciousness regarding these concerns, for which the rates and chances of miscarriages have been rising these days. Besides, a pregnant woman might face many major or minor diseases or disorders which they might not be familiar with, and hence need a doctor's consultant to deal with this. However, availability of a doctor might be challenging sometimes, since time management is an issue in this case; also visit to doctor can prove costly. Fortunately, these issues can now be overcome with the help of technology, since the enhancement of technology is playing a vital role in changing the world digitally. This increase in the technology and availability of the internet is causing an increase in the usage of smart phones among people. Since more people are now accustomed to smart phones, the systems are now able to meet various requirements of a pregnant woman for which assisted living in pregnancy can now be served at its best.

Our research is based on suggesting a proposal on the development of an intelligent and smart application, that aims to conduct operations based on diagnosing the complications the mother faces during her gestation period as well as developing a friendly chatbot to cater to the user's mental needs. The system utilizes a collection of the previous records of a patient's medical report, diagnosis, doctor's suggestions, dietitian's sayings, related exercise tips, and some other necessary details to treat the patients both physically and mentally by implementing machine learning.

1.2 Research Problem

In today's society, a smartphone has become a necessary item in people's lives. Ordinary people rely on their phones and internet for any small inconvenience they face in their day to day life. Even in regard to major complications, many people still rely on their internet for at least gaining a general idea about the problem. Similarly, this is very common in pregnant women as well. T. Peyton [14] et. al., (2014) conducted a research where most of the women reported that their first prenatal visit only took place after the end of the first trimester. Instead of visiting the doctors, they relied on internet searches to determine whether or not they are pregnant. Moreover, during their second trimester, when they receive printed literature from their first medical visit which they referred to as "useless" or "overwhelming" amount and they once again turn to "Dr. Google" or mobile apps. One of the participants reported that her reason for using "Dr. Google" instead of an actual Doctor, is because she could not get an appointment until she was at 14 weeks. And, such experiences are more common among the people residing in rural areas. Hence, for reasons such as these, pregnant women find help, support and information from the Internet and commercial mobile applications. And, as technology and its availability advances, such reliance on it is only going to increase. Therefore, in order to support pregnant women in their initial trimesters and provide them with necessary guidelines, an online system is the most viable option in today's time. But this system needs to be more approachable to pregnant women and catering to their needs. There are already many apps for pregnancy but not all of them are effective. According to [13], many pregnant women consider their pregnancy to be unique and different from others which leads them to neglect many apps or its suggestions as they think that their situations are not the same. Furthermore, pregnant women limit their sharing of pregnancy to their immediate family and often do not like to share outside of it. So, the system needs to come off as very privacy respecting to them. Last but not least, physical health is not the only main concern for pregnant women as many women suffer from mental health issues during pregnancy which can also lead to many problems as well as in worst case, pregnancy loss.

Therefore, the aim of our research is to answer the question,

How can we build an effective system that caters to both the physical and mental health of a pregnant woman while making the experience unique as well as engaging?

This research will answer the above question by testing various machine learning models and finding the suitable one to create a mobile application that predicts the condition of pregnant women as well as creating a friendly chat bot to cater to the user's mental needs.

1.3 Research Objectives

At different stages of pregnancy, women can face several symptoms which they might not be familiar with. Besides, it is usual that most of them lack prior knowledge regarding the fetus development or any major and minor complications that they might face or need to deal with, which results in lack of consciousness about their health and hence, may lead to serious affliction. Moreover, visiting doctors for any minor causes can be difficult for a pregnant woman. Therefore, these issues could have been minimised, if they were guided regularly and treated immediately without taking the hassle of leaving home. In light of all these circumstances, in developing countries such as Bangladesh, such a system is required that can help pregnant women deal with these situations at different stages of pregnancy [19]. This has encouraged us to develop a system that can help in guiding the expecting mothers. Related features and suitable keywords will be presented to the user while providing the inputs. Besides,language preferences will be given, so that the user can chat and share their thoughts in an ease.

The objectives behind suggesting to serve this assistant to pregnant women are :

1. To provide constant assistance to pregnant mothers for both physical and mental health complications.

2. To limit the number of prenatal visits for minor concerns.

3. Serving the individuals with proper nutrition guidelines to avoid malnutrition and other health complications.

4. Presenting a user-friendly environment so that people living in rural areas can find it easy to use the app and get assisted instantly.

5. To notify and alert the person much earlier to take precautions before facing any serious disorder.

Chapter 2

Literature Review

Considering all the difficulties a pregnant woman may face, certain systems and measures have been previously proposed by many expert researchers. S.N. Tumpa [19] et. al., 2017 proposed a smart mobile and web application system which is an automated maternity helping aid to pregnant women of Bangladesh. The system takes all necessary information from the pregnant woman which is stored in a database by cloud manager along with the details of their preference and allergy issues and food items to serve perfect nutrition guidelines. Besides, it calculates the due date, presents a graphical representation of the condition of the fetus, provides necessary advice to keep the mother and unborn child healthy as well as notifies the user and her spouse earlier about the regular checkup. Moreover, the system allows communication with experts and researchers to prepare nutrition guidelines. If the system detects that the user's health condition is at risk, it will quickly notify them and their close relatives via SMS to take required steps. Besides, it sends reports to the experts for guidelines and the Global Positioning System (GPS) is used to find out the nearest clinic's location to consult doctors immediately. G. Saranya [17] et. al. developed a system called e-antenatal assistance which provides guidance to the pregnant women throughout the course of pregnancy to prevent potential health problems while promoting a healthy lifestyle. This mobile and web-based application gathers information from pregnant women using surveys and questionnaires and provides diagnosis to health issues, diet tips etc. with supervised machine learning techniques, decision tree analytics and cluster analytics. It also acts as a repository for all the medical history, reports, diagnosis provided and location of the patient. The accuracy can be improved more by continuous learning and having more diagnosis data over time. Until then, doctor's supervision is needed manually to correct the diagnosis result. W. L. Moreira [16] et. al., 2016 presents a mobile monitoring solution to identify worsens in the health status of pregnant women suffering from hypertensive disorders using Naive Bayes Classifier and help experts in decision making. To measure the risk factors, physiological data are collected from the patient using the sensor in the system and other information like protein in urine is checked by performing a proteinuria test and then calculates the probability of the pregnant woman having changed her condition with an accuracy of around 84 percent. A. B. Queyam [24] et. al. developed a hardware prototype named "Feto Maternal Care Unit (FMCU)" that consists of several biomedical sensors attached to the mother's body which collects bio-physiological signals like maternal respiration rate, body temperature, fetal movement, abdominal ECG, etc. For real time signal visualization, processing and data logging the national instruments (NI) myRIO transmits physiological signals wirelessly to a PC running NI LABVIEW software. For processing, decomposition and feature extraction, various signal processing algorithms like EMD, VMD, EWT etc are used. An Android-based smartphone application was also developed that sync data directly to the cloud which can be accessed by an authorized doctor for further supervision and diagnosis. N. Yalina [21] et. al. developed an early warning information system that can detect pregnancy risks so that pregnant women and health professionals could anticipate it and take actions at an early stage. They developed an application for the website which was built with RWD, created the database where the information of the pregnant woman will be stored and later used it to calculate early pregnancy risk score based on Poedji Rochjati Scorecard and provide recommendations.

On the other hand, A. Fourney [15] et. al., 2015 used search query logs of pregnant women to learn a detailed model of the phenomenon by aligning search queries with the 40 gestational weeks of pregnancy to show through detailed log analysis that information needs of new and expectant mothers wax and wane in predictable patterns over the course of the weeks of pregnancy and beyond. Leveraging these temporal patterns to anticipate the queries of new parents will allow the system to better meet the information needs of these people as well as can be used to investigate public health concerns. The model they constructed could also classify searchers as pregnant, non-pregnant or new mothers and align their histories with gestational weeks to estimate due dates or delivery dates. After comparing distribution of births by gestational week as estimated by this model and by CDO, Pearson correlation of $r=0.97$ ($p>0.0001$) was found though the model skewed slightly due to systematic noise in data.

There are a number of sides of pregnancy but the mHealth Design focuses mainly on health issues. T. Peyton [12] created a structuring health concept that she calls the "pregnancy ecology", accounting for the multi-faceted experience of pregnancy as a transformational event. After conducting some interviews, she came to the conclusion that the pregnancy ecology has 5 facets -medical, social, informational, technical and intangible. Thus mHealth designs should work on holistic health, social and overall life support of the pregnant women rather than just providing medical guidelines for physical health. On another paper, T. Peyton [13] et. al.,2014 conducted a design-oriented qualitative study on lower-income pregnant women in Pennsylvania and presented the needs that have to be accounted for mHealth design which includes first trimester self-guided information seeking, using immediate needs as a hook for long-term concerns, including some (but not all) of the social circle and tailoring to the uniqueness of each woman's pregnancy.

Being illiterate and far from low socio-economic status, indigenous women are not able to fully enjoy the blessings of technology. N. Bagalkot [22] et. al. reported initial engagement with the community stakeholders for pregnancy care in south India. They also found tension between traditional and requirements for modern pregnancy care, lack of coordination between multiple stakeholders in pregnancy care and the role of physical and digital infrastructure in pregnancy care. According to N. Verdezoto [30] et. al. women in rural Ecuador are experiencing extreme poverty, malnutrition, ethnic and linguistic differences, poor access to basic resources, education and health services that is not only exposing them to discrimination but also putting them at risk of serious pregnancy complications. After conducting some focus group discussions, the researchers identified some barriers to antenatal care such as unknown terminologies used during consultation, referral process, long waiting times for consultations, missing appointments, effective dimensions of the healthcare settings etc. Thus, the study revealed that the potential use of mobile technologies, simple interventions such as apps with translations of medical terminologies and self-monitoring systems can provide support and medical care to indigenous women. In both articles, they tried to address the concerns and highlight opportunities regarding pregnancy in rural areas to further develop technology designs that solves these issues.

From the above discussion, it is observed that multiple systems have already been developed to serve pregnant women during her gestational period to support her with proper medical care. However, it is seen that there are still a lot of aspects that need to be considered while developing an mHealth system that can completely fulfill the needs of pregnant women of every class which makes this side of the research promising.

Chapter 3

Methodology

3.1 Methodology for predicting pregnancy outcome

To make our system smart enough, our first step would be to make the system learn new things in order to cope with the environment. To make this happen, the system needs to be fed with data based on which it will respond to the input and for this, it is necessary to train and predict the performance of the collected data. The proposed model assists to differentiate among various pregnancy stages. The model uses raw data where feature selection techniques are being applied to collect more manageable groups for processing. These groups of data are then trained and tested with various models to determine and predict the percentage accuracy of each feature which defines the chances of occurrence of risk factors in a pregnant woman. The system aims to identify the possibility of having preterm or full term birth,the chances of c-section or vaginal delivery, number of child births, jaundice and other related features with the data of mother's health features as inputs.

The model maintained the following stages shown in **Figure 3.1** .

1. **Input Data:** The dataset containing the health related datas and features of a pregnant woman are arranged and structured, in this stage, to be used by the model for preprocessing.

2. **Data pre-processing:** The features containing the missing data or null values are handled in this stage by both filling the space with the max occurrence values and dropping the columns with less relatable features, preparing the dataset to be used for the next stage.

3. **Feature Selection:** The necessary features are both manually and randomly chosen, in this stage, using some feature selection techniques such as Univariate Selection, ExtraTreesClassifier, Correlation Matrix with Heatmap which provides a visual representation of K-best features which contributes most to our prediction variable or output in which we are interested in. This needs to be done since irrelevant features can decrease the accuracy of the models and make the models learn based on irrelevant features.

4. **Classification:** In this stage, a small percentage of data say around 20-30 percent are organised and classified by relevant categories so that it can be

used more efficiently. The data are trained with various classification models and tested for accuracy of each predicted variable and the output determines whether the patient would be at risk and helps to implement controls to mitigate risks.

5. **Fine Tuning:** This stage involves the model parameters to be optimized or tuned by the training process. Here, we run the data through the operations of the model, compare the resulting prediction with the actual value for each data instance, evaluate the accuracy and adjust until we find the best values. Hyper parameters are tuned by running the training process, looking at the aggregate accuracy and adjusting to find the best combination to handle our problem.

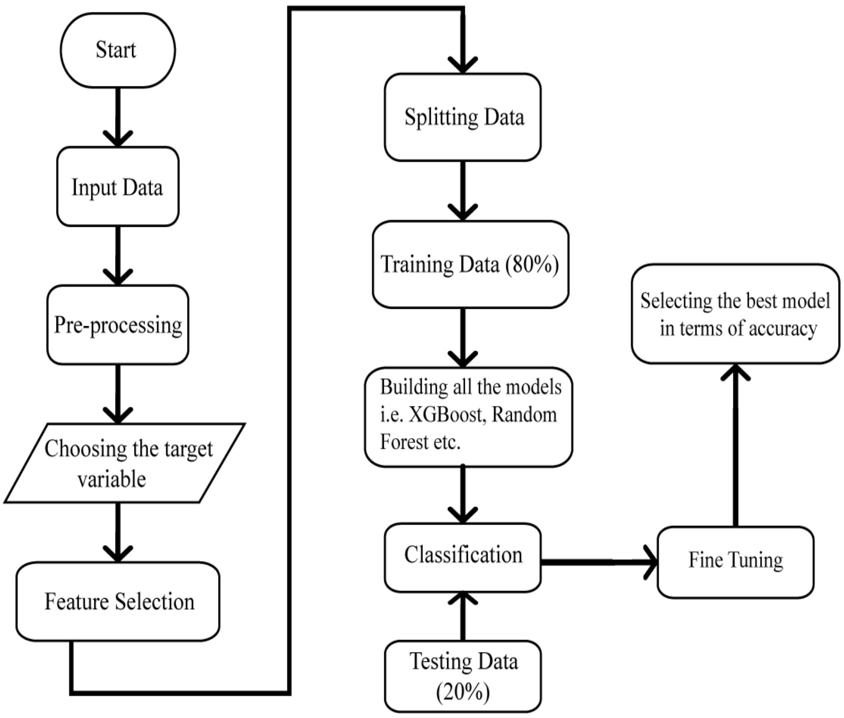

Figure 3.1: **Flowchart of the proposed model**

3.2 Methodology for Question Answering Chatbot

To provide an easier, engaging and approachable experience, a question answering chatbot is the best solution that we could find. For this, we are following the architecture made by Sunil Jammalamadaka to create a chatbot using BioBERT and GPT2 [33]. Since, BioBERT specializes on biomedical domain, we deemed it perfect for our work. Let us look at the architecture shown in **Figure 3.2**:

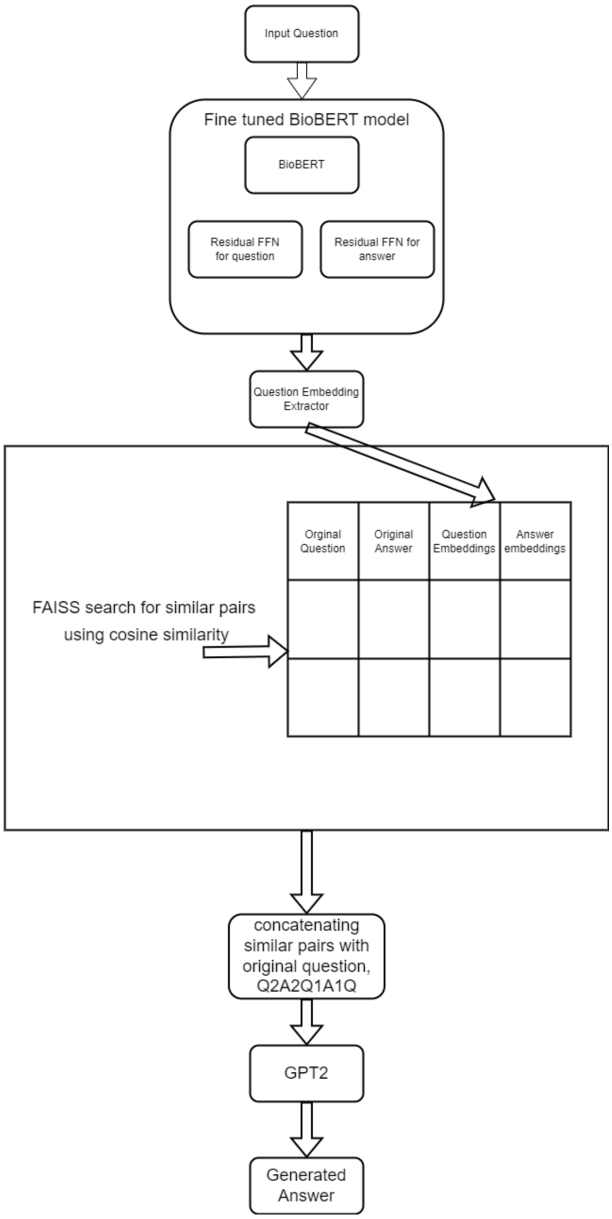

Figure 3.2: **Inference pipeline for chatbot**

When the user asks a question, the chatbot will take it as input and pass it through the fine-tuned BioBERT model and with the help of question and answer embedding extractor, we extract embeddings for FAISS search to find similar pairs using cosine similarities. Then, we concatenate similar ones with the original in order and pass it through GPT2 to use its generative modeling to finally generate an answer.

Chapter 4

Model Specification

4.1 Machine Learning Based Classification

Since our dataset has a large amount of features with different data types as well as multiple target outcomes, we have decided to use multiple classifiers suitable for each of the target outcomes to conduct our research. The classifiers that we selected for our research are: DTC, RFC, KNN, SVM, Logistic Regression, Naive Bayes and XGBoost.

Decision Tree Classifier is a frequently used classifier in classical machine learning approaches. It is a tree where each node shows a feature (attribute), each branch shows a decision (rule) and each leaf shows an outcome (categorical or continuous value). [23] As decision trees resemble human level thinking, it is an easy task to grab the data and make some good interpretations. It considers all possible alternatives and traces each of them to come to a conclusion in order to make easy comparisons among the alternatives. One of the best advantages of DTC lies in its transparency as well as the ability to select the most biased feature and comprehensibility nature.

Random Forest Classifier is quite similar to DTC but instead of using one decision tree, it constructs a tree randomly from a set of possible trees with each of them having K random features at each node. Here, "randomly" means that each tree in the set of trees has an equal chance of being sampled. [8] shows that RFC achieves increased performance and generates accurate and precise results if it is given a large number of instances.

K-Nearest Neighbour Classification is a very straightforward supervised machine learning algorithm which is used to solve both classification and regression problems. It makes the assumption that similar things exist in close proximity and predicts the class based on that. [31] Thus, it has two stages: the first is the determination of the nearest neighbours and the second is the determination of the class using those neighbours. The K in KNN is the parameter that the classifier takes along with the data. The K is the number of neighbours to consider when classifying.

Logistic Regression is the most famous machine learning algorithm right after linear regression. While linear regression algorithms are used to predict values, logistic regression algorithms are used for classification.[6] The natural logarithm of an odds ratio: logit is the underlying mathematical concept of logistic regression. It can be a powerful analytical technique for use when the outcome variable is dichotomous.

Support vector machine is a popular algorithm as it generates significant accuracy with little computation power. SVM finds a hyperplane in an N-dimensional space (N - the number of features) which distinctly classifies the data points. Here, hyperplanes are decision boundaries that aid in classifying the data points and support vectors are data points that are in close proximity of the hyperplane and influence the position and orientation of the hyperplane. [3] SVM has been successfully used for medical diagnosis.

Naive Bayes Classifier is a probabilistic machine learning model which can be used for classification tasks. The core of the model is based on the Bayes' theorem. Using Bayes theorem we can find the probability of event A happening, given that event B has already occurred. The classifier uses this theorem to utilize the given features to predict the target outcome. [5] Even though it makes some unrealistic independence assumption, the classifier is shockingly effective in practice as its classification decision may often be correct even if its probability estimates are inaccurate. And, it works best in two cases: completely independent features and functionally dependent features. Although, it performs its worst between these extremes.

[26] Compared to others, **XGBoost** is a fairly recent machine learning model that is highly preferred by many because of its good accuracy and efficiency out of core ensemble learning. The base learner of XGBoost is a decision tree and it uses a generalization of the gradient boosting method for the construction of the next base learners. It is quite different from the other ensemble learner random forest. Furthermore, it is also the sparsity aware algorithm which has the ability to handle the null values of historical data. The missing values of the historical data can be processed without any imputation preprocessing.

4.2 BioBERT

Before we talk about BioBERT, let us look at the model that it is based on, BERT. BERT [25] is a contextualized representational model that is formulated on a masked language model and pretrained using bidirectional transformers [20]. Here, the transformer architecture is basically a stack of six encoders and decoders where each encoder consists of self attention layer and feed forward network. These two layers help the encoder to understand the context. Whereas, each decoder has three layers: self-attention, encoder-decoder attention and feed forward. The purpose of decoder is to map one language with another.

And, the BERT architecture is a stack of these encoder blocks of transformer architecture which uses bidirectional approach (**Figure 4.1**). While previous language models were limited to a combination of two unidirectional language models, BERT incorporates a masked language model that predicts randomly masked words in a sequence by using bidirectional encoder transformer which reads entire sequence of words at once. With this exceptional architecture, it manages to achieve state of the art performance on most NLP tasks without requiring too many task specific modifications [29].

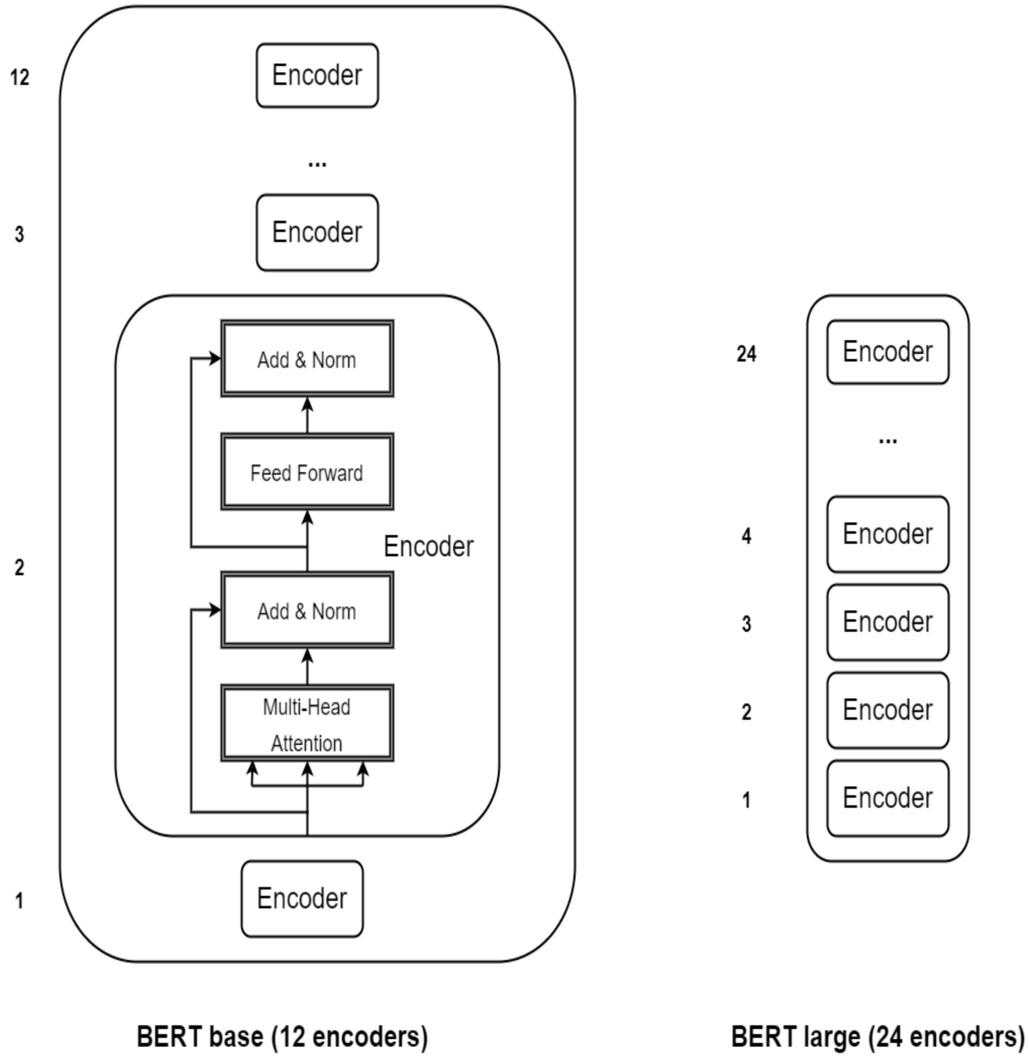

Figure 4.1: **Architecture of BERT**

Finally, BioBERT is a pre-trained language representation model that is based on BERT and specifically for the biomedical domain. With weights from BERT, BioBERT is pre-trained on biomedical domain corpora (PubMed abstracts and PMC full-text articles) and fine-tuned on three popular biomedical text mining tasks which are Named Entity Recognition, Relation Extraction and Question Answering. On biomedical QA datasets, all versions of BioBERT significantly outperforms BERT and other state of the art models, specifically BioBERT v1.1 which uses PubMed corpus, achieves a strict accuracy [29].

As, our goal is to find a way to build an AI that will help answering pregnant women's queries, BioBERT is the best possible model for us to perform Question Answering tasks on dataset that contains medical information for pregnant women.

4.3 GPT-2

Another state-of-the-art model Generative Pre-trained Transformer 2 or what we know as GPT-2, will be serving as another integral part of our chatbot. This high-quality advanced text generator does the task by iteratively predicting the next token in the sequence after pretraining the transformer decoder architecture (**Figure 4.2**). A. Radford [27] et. al., 2019 intended to build a general system that can perform various tasks without the need to manually create and label a training dataset for each one. Thus they upgraded their original supervised learning-based GPT to a large transformer-based language model with 1.5 billion parameters, 48 layers, 1600 states dimension, 1024 context token size, and 512 batch sizes that can learn to perform various Natural Language Processing tasks, such as question answering, reading comprehension, machine translation, and summarization without explicit supervision when trained on new datasets. GPT-2 is even said to outperform models that were trained on some domain-specific datasets.

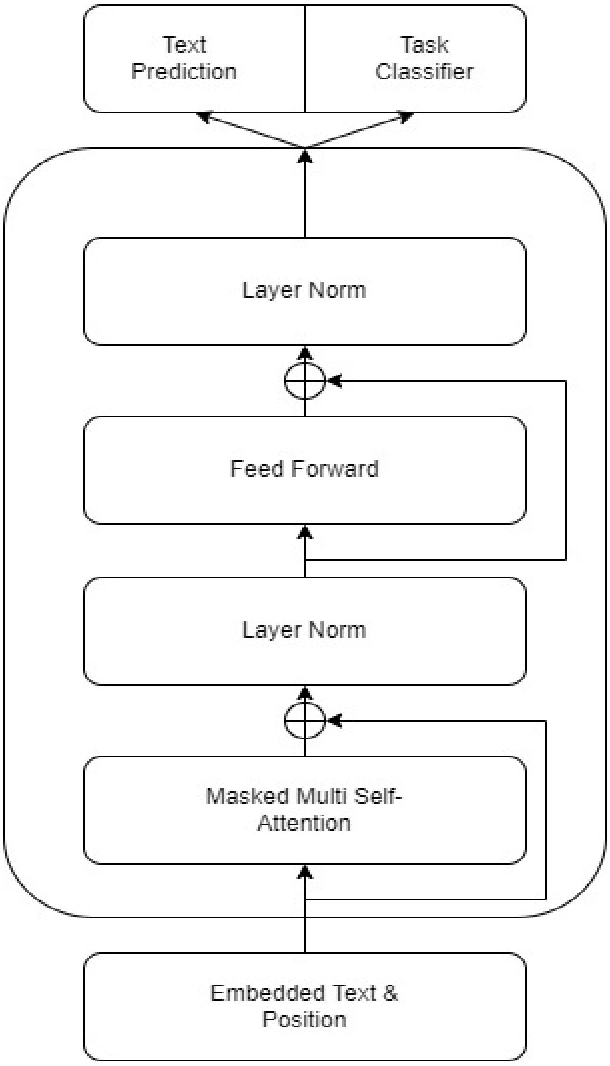

Figure 4.2: **Architecture of GPT-2 model**

GPT-2 model works as a zero-shot learner as it's trained on unlabeled data and in some cases, it outperforms supervised learning too. The prime objective of this model is to predict the next probable word given all of the preceding words in a document. GPT-2 can be used to generate conditional synthetic text samples and we then let it generate a lengthy continuation of the input provided.

According to F. Jelinek [1] et. al., 1980 and Y. Bengio [7] et. al., 2003 because of the language natural sequence order, the joint probabilities over symbols will be factorized as the product of conditional probabilities where for this unsupervised distribution estimation, $(x_1, x_2, ..., x_n)$ are a set of examples each composed of variable length sequences of symbols $(s_1, s_2, ..., s_n)$

$$p(x) = \prod_{i=1}^{n} p(s_n | s_1, ..., s_{n-1}) \tag{4.1}$$

This equation of **(4.1)** allows for tractable sampling from and estimation of $p(x)$ as well as any conditionals of the form $p(s_{n-k}, ..., s_n | s_1, ..., s_{n-k-1})$.

Since a general system has to be capable of performing many different tasks, it should condition on both the input and the task to be performed and can be expressed in a probabilistic framework as estimating a conditional distribution $p(output|input, task)$ whereas for a single task it would have been $p(output|input)$.

Chapter 5

Predicting pregnancy outcome using classification models

5.1 Dataset description

The scarcity of datasets related to pregnancy posed a great difficulty for us. But eventually we were able to find a suitable dataset for our research problem. The dataset known as " Mother's Significant Feature (MSF)" was collected from the Mumbai metropolitan region in Maharashtra, India. The women from whom the data was collected were interviewed just after childbirth between February 2018 to March 2021. MSF consists of 450 records with a total of 130 attributes consisting of mother's features, father's features and health outcomes. All the features are possible complications associated with child health, mother's health and health outcome. And, these were selected after detailed literature review and brainstorming sessions with doctors (gynaecologist and paediatricians).

The data types of the attributes vary a lot i.e. attributes like PCOS (Polycystic ovary syndrome) and Fertility treatment have binary datas with 1 indicating the subjects have them and 0 indicating they do not whereas attributes such as Age of Mother, Height have integer data types and Hemoglobin, weight of baby have float data types. A major complication of the dataset is that more than half of the subjects have no inputs in some attributes such as weight before delivery, number of siblings, Miscarriage history etc. Moreover, many attributes have sub parts of their own to further specify information. For example, the Menstrual Cycle is divided into two parts: Before Marriage and After Marriage. Attributes like Diet and sleep pattern have multiple parts specifying what sort of diet or during which stage of their lives.

It is also worth mentioning that the "Mother's Significant Feature (MSF)" dataset has been used in other research papers that worked on Preterm birth and NICU prediction. Using the same dataset, [32] identified that health conditions like diabetes and hypertension have an influence on pregnancy outcome while lifestyle habits such as consumption of alcohol, unhealthy food also affects the pregnancy outcomes specified as preterm birth and NICU requirement.

5.2 Dataset Visualization

In this section, multiple pairplots are being presented to show the correlation among the features for each of the predicted variables. Among the various extracted features, we have selected three random features which may create an impact or bring a change in the accuracy score. The correlation of these three features are then compared to the predicted outputs such as, preterm or full term birth, c-section, vaginal delivery, jaundice, NICU Stay, and some other output variables which are then presented graphically as shown in **Figure 5.1**:

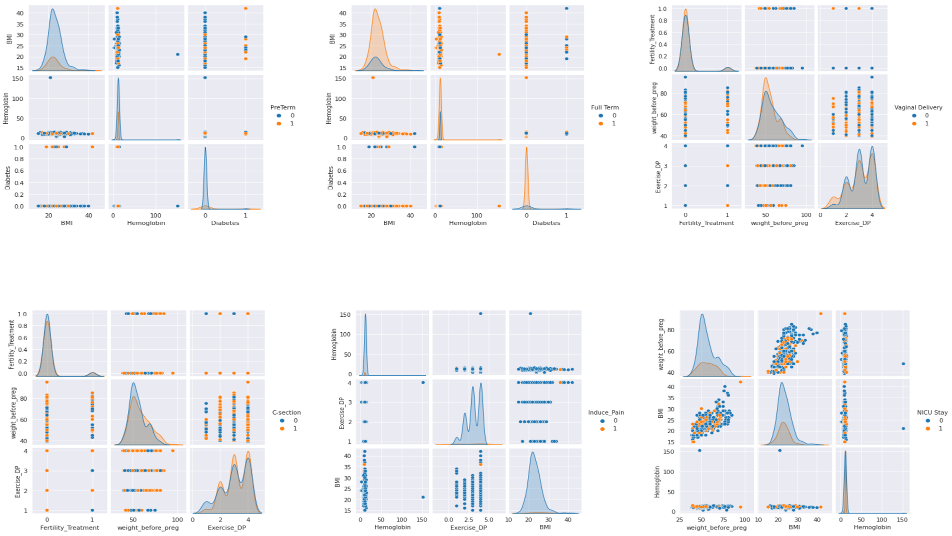

Figure 5.1: **Pairplot graphs representing the relationship among the extracted features and predicted output**

In **Figure 5.1**, multiple pairplot graphs are shown where three of the more relevant extracted features are displayed respective to the predicted output. The scattered points on the figure represent independent features that provide different information. An example can be the relationship between haemoglobin and diabetes for the preterm, where we can see that the points are scattered and independant. The other figures, however, show either a linear or non-linear relationship. An example can be the relationship between BMI and weight before pregnancy for the NICU Stay which shows a linear pattern and is kind of dependent on each other.

5.3 Dataset Preprocessing

As it has been mentioned earlier, many of the attributes have sub parts specifying certain context. In the dataset file, they each had their own columns but under one label. To make it easier for our program to read the dataset, we turned all these sub parts into their own individual parts while renaming them like "Menstrual Cycle_M" and "Menstrual Cycle_AM" specifying the times before marriage and after marriage.

Since, more than half of the subjects have no input in some certain attributes, the best possible solution that we could find to this missing value problem is to delete those attributes. With such a high rate of missing values those attributes will not be of any use and most likely will hamper the results. Moreover, there were some duplicate features which we also chose to remove from the dataset. Along with that, we deleted the Mother_UID column as they are just numbers to identify each subject and do not indicate any features.

5.4 Primary Data Analysis

5.4.1 Feature Selection

Although many features are generally seen to be used in data representation, very few features may actually contribute to reaching the goal as most of them might not be related to the target outcome [2]. The performance of the model can be negatively affected by the presence of these irreverent and insignificant features, adding uncertainty to the predictions and reducing the overall effectiveness of the model. Thus feature selection becomes vital before training the model as it can reduce overfitting and training time along with improving the accuracy. Feature selection can be distinguished into unsupervised and supervised methods where supervised mechanisms can be done in 3 different ways: Wrapper, Filter and Intrinsic.

Statistical measures provided by scikit-learn library such as Pearson's Correlation Coefficient, ANOVA, Chi-Squared, Mutual Information etc. can be used for determining the correlation. Once statistical correlations have been calculated for each input variable with the target, filtering methods provided by scikit-learn such as the two most popular methods SelectKBest or SelectPercentile can be applied.

Three types of feature selection techniques were used for the purpose of this paper which are Univariate Selection, Feature Importance and Correlation Matrix with Heatmap. All these techniques are considered to be very popular as they perform really well.

1. **Univariate Selection**: This feature selection method performs univariate statistical tests to examine each feature individually and determine the strength of the relationship of that particular feature with the targeted one. The feature selection is then done by selecting the ones with the best result. Univariate Selection is generally good for gaining a better understanding of the data.

   The statistical analysis can be done using Pearson's Correlation Coefficient, Chi-Squared, ANOVA, Distance correlation, Mutual Information and Maximal Information Coefficient (MIC). The feature selection can be carried out using SelectKBest, SelectPercentile, GenericUnivariateSelect etc. In this paper, we will be using the Chi-Squared test for calculating the statistical score and SelectKBest to select the best possible k features.

   Chi-Square is usually used when on categorical data. The formula **(5.1)** is used to calculate the Chi-Squared.

$$
\begin{aligned}
x^2 &= \frac{(O_{11}-E_{11})^2}{E_{11}} + \frac{(O_{12}-E_{12})^2}{E_{12}} + ... + \frac{(O_{mn}-E_{mn})^2}{E_{mn}} \\
&= \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{(O_{ij}-E_{ij})^2}{E_{ij}}
\end{aligned}
\tag{5.1}
$$

X^2 = Chi Squared

$\sum$ = The sum of

O = Observed value

E = Expected value

Here, "O" stands for observed value and "E" stands for expected value if these two categories are independent. The Chi-squared value will be high if O and E values are related but the values of O and E will be close in case they are independent.

2. **Feature Importance**: A method of assigning a score to input features based on its usefulness in predicting the target is known as feature importance. It is generally calculated through statistical correlation scores, decision trees, and permutation importance scores or coefficients calculated as part of linear models. This feature importance score can not only be used in reducing the number of input features but also gives us a better understanding of the data and the model as well. Tree based embedded feature selection algorithms are commonly seen used in machine learning and data mining as they do a very decent job [18]. For the feature importance in our paper, we will be using an inbuilt class that comes with a Tree Based Classifiers, Extra Tree Classifier, for extracting the best features from our dataset.

3. **Correlation Matrix with Heatmap**: Table which represents the correlation between the features is known as the correlation matrix. This plays a vital role in helping to visualize the correlation between all the possible pairs of values in a table. The matrix consists of rows and columns and each cell in the correlation matrix contains the calculated correlation coefficient for the pair. There can be three possible outcomes of the correlation: a positive correlation, a negative correlation, or no correlation. Positive correlation will mean increase in one value of feature causes increase in the value of the other variable whereas negative correlation will mean increase in one value of feature causes decrease in the value of the other variable. No correlation depicts that the variable doesn't affect the other variable at all. In this paper, we used heatmaps of correlated features to visualize the correlation between the features and the target. The heatmaps were generated using the seaborn library.

By applying different algorithms on the data the final features can be selected considering the classification performance [10]. Thus we performed all the above three feature selection techniques and determined the final selected features that give us the best accuracy.

5.4.2 Visualization

When training the datas with various classification models, it is necessary to extract relevant features to come up with the best accuracy result after the test of each predicted variable. Feature extraction techniques play a great role in this case, to serve us with necessary features needed for the test. However, it is important to understand how relatable and connected are each of these extracted features to the predicted variable so that the outcomes generated are quite relevant and accurate. For this, the correlation among each feature is needed to be found and since visualization of data makes it easier to understand the relationship among datas, different visualization methods have been used so that it becomes easier to detect patterns, trends and outliers in groups of data.

The visualization technique that has been used to understand the correlation between each and every independent feature and the output is the Correlation Matrix with Heatmap. The seaborn library is used to generate this heatmap, where a fixed size of data is set to the matrix and the result generates the correlation value between 0 to 1 for each feature. To understand the relationship, the output feature is compared to the correlation value of the independent features, where the greater values closer to 1 defines a strong correlation with darker colors. Thus the higher value features are correlated. Using this technique, we can now choose and understand what are the features that are very important with respect to the target output. This way, the features with correlation value greater than 0.2 are used in particular model training.

After extracting the correlated features, different classification techniques have been applied where we trained and tested the accuracy of the data and this has been presented with a visualization plot named Receiver Operating Characteristics (ROC) curve. The ROC curve displays the performance of our predicted variable or output which are binary classifiers. It graphically represents the True Positive Rate (TPR) versus False Positive Rate (FPR) at various classification thresholds from 0 to 1 that best suits our application and helps to maximize correct classification or detection while minimizing false positives. The closer the curve gets to the upper-left hand corner, the better the model is. Later, the AUC (area under the curve) score is calculated which displays the probability that the model ranks a random positive example more highly than a random negative example.

5.5 Implementation

Python is used to implement and test the proposed model for Pregnancy risk detection. Because of its simplicity, easy to use architecture and a wide variety of libraries, python is the go to language for machine learning. Libraries like numpy and pandas make it easy to take the input data file in the form of a .csv file and convert it into a dataframe. All of the data pre-processing including imputation can be done in python and the sklearn library provides the necessary models for data scaling and such. Furthermore, the sklearn library also provides all the models that we have selected to implement and test. Along with that, seaborn and matplotlib libraries can help to visualize the correlation between features by generating heat maps and plotting graphs, thus making the feature selection process easier.

5.6 Statistical Relationship

For a dataset with 135 columns, feature selection became a vital part for the purpose of this research. While performing the three feature selection techniques to understand the relationship between the features, we came across the fact that most of the features were redundant and thus misleading the research providing a poor accuracy. Out of the three methods, applying feature importance with Extra Tree Classifier was performing well as the selected features based on the feature importance were proving to improve the accuracy of the models much higher than the others. So we primarily selected the best features provided by the Extra Tree Classifier and then updated it according to the features chosen by the SelectKBest and observed the correlation heatmap. Top selected important features per target variable of our research are provided in **Figures 5.2 to 5.7**.

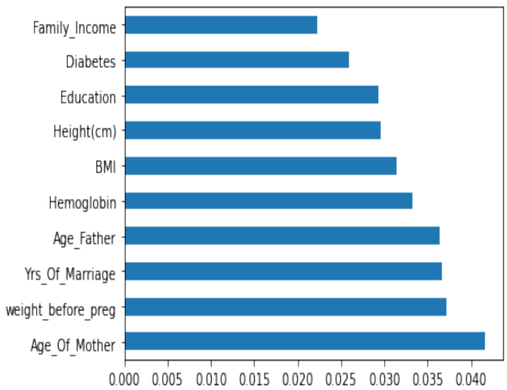

Figure 5.2: **Top selected features for "PreTerm" using Extra Tree Classifier.**

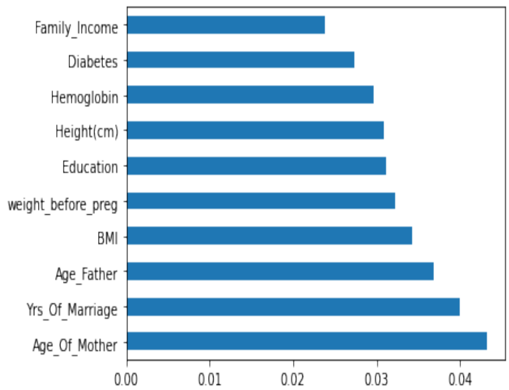

Figure 5.3: **Top selected features for "Full Term" using Extra Tree Classifier.**

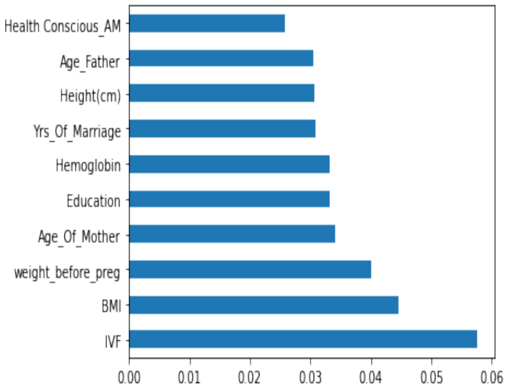

Figure 5.4: **Top selected features for "no of births(single/Twins)" using Extra Tree Classifier.**

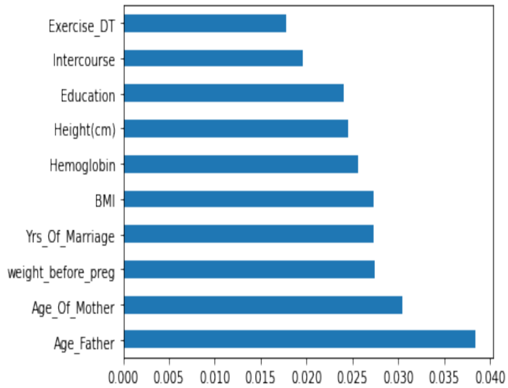

Figure 5.5: **Top selected features for "NICU Stay" using Extra Tree Classifier.**

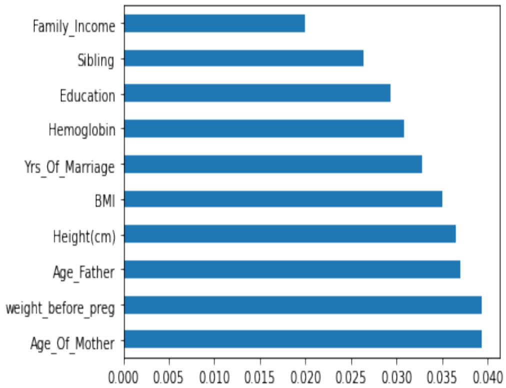

Figure 5.6: **Top selected features for "C-section" using Extra Tree Classifier.**

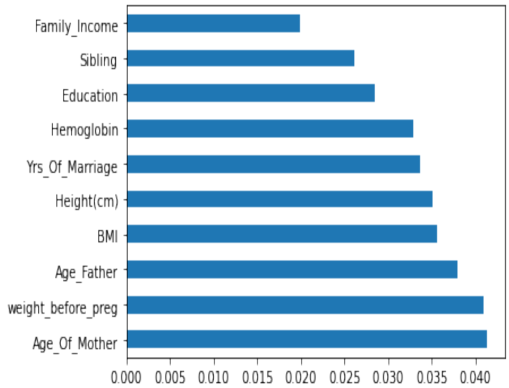

Figure 5.7: **Top selected features for "Vaginal Delivery" using Extra Tree Classifier.**

In **Figure 5.2**, we can see that Extra Tree Classifier depicts that features like "Age Of Mother", "BMI", "Hemoglobin", "Height(cm)", "weight before preg", "Diabetes", "Family Income", "Yrs Of Marriage", "Education" etc are the most important features in predicting preterm birth and selecting these features really did improve the accuracy. Another important point to note is that features like "Diabetes" and "Hemoglobin" drastically sent the accuracy higher conveying that these are very crucial in determining preterm labor possibility and even in real life pregnant women with pregestational diabetes mellitus are proved to have higher chance of going through preterm labor [4]. This gives us a glimmer of hope about a great opportunity that the selected features can be researched further to understand why features like "Education" and "Family Income" seem to have such high correlation with pregnant women giving preterm birth.

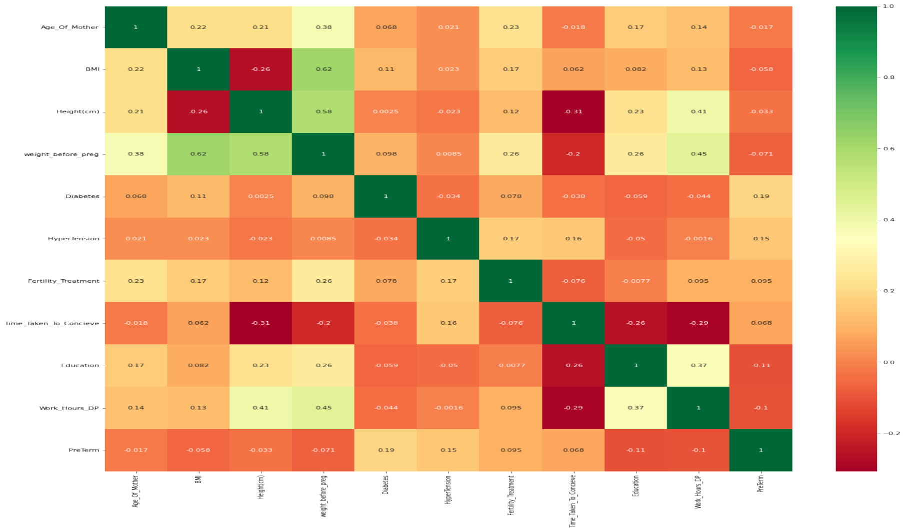

Figure 5.8: **Correlation matrix with heatmap for "PreTerm."**

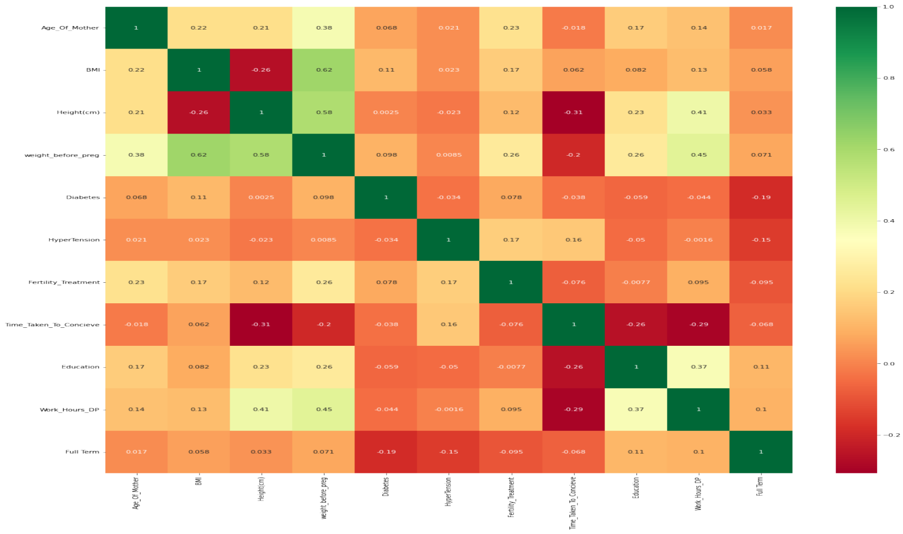

Figure 5.9: **Correlation matrix with heatmap for "Full Term."**

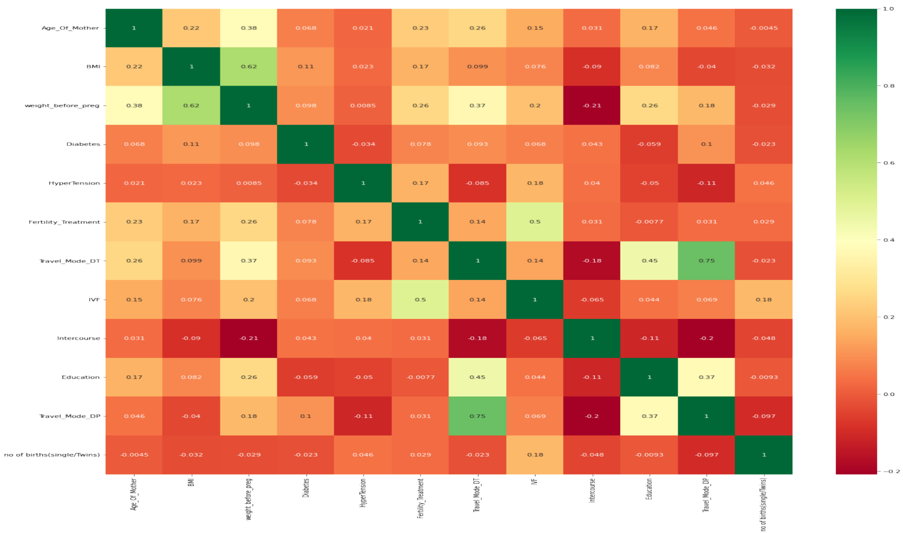

Figure 5.10: **Correlation matrix with heatmap for "no. of births (Single/Twin)."**

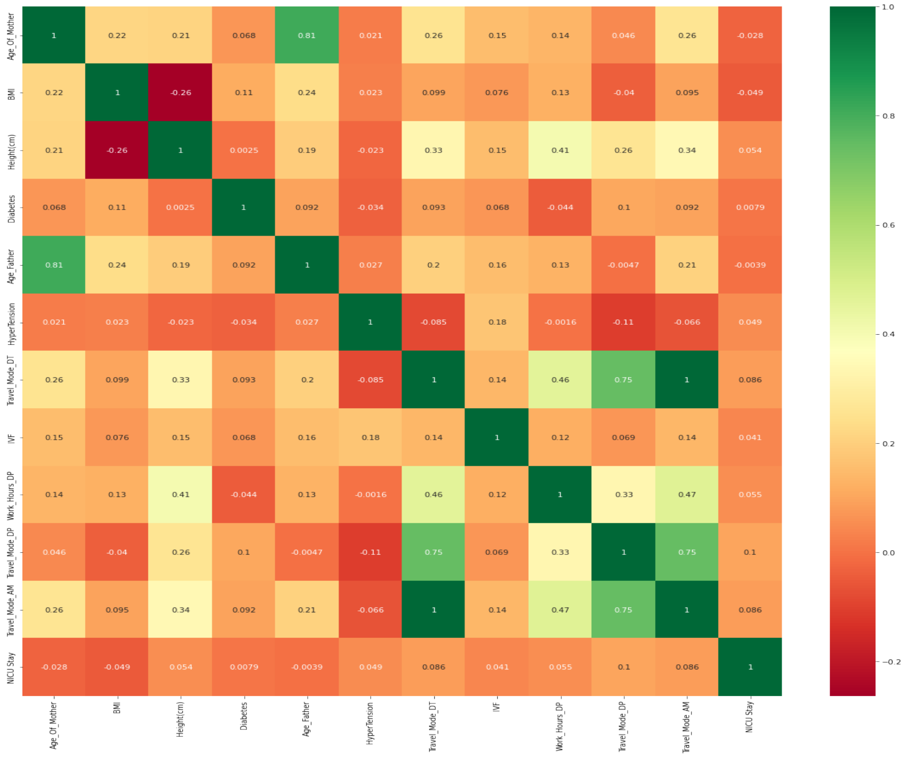

Figure 5.11: **Correlation matrix with heatmap for “NICU Stay.”**

Figure 5.12: **Correlation matrix with heatmap for “Jaundice.”**

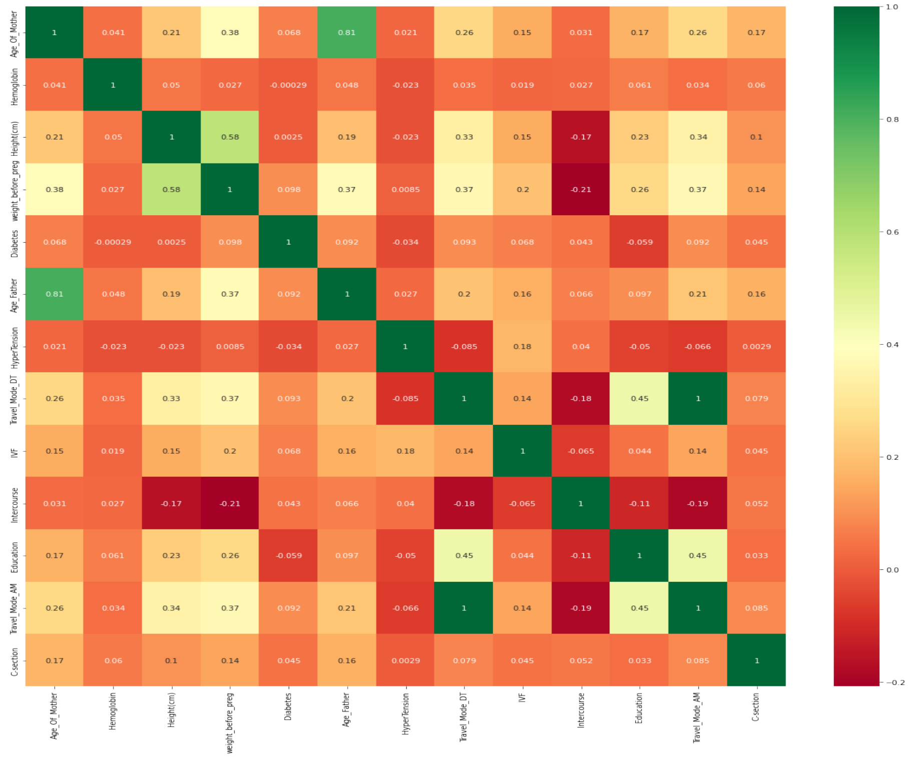

Figure 5.13: **Correlation matrix with heatmap for “C-section.”**

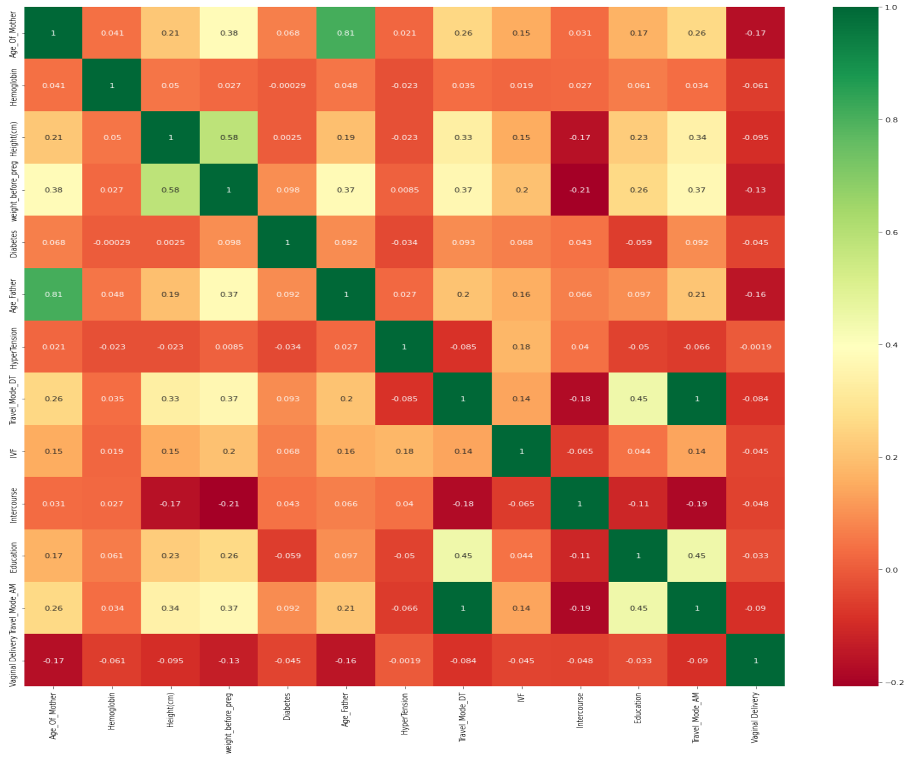

Figure 5.14: **Correlation matrix with heatmap for “Vaginal Delivery.”**

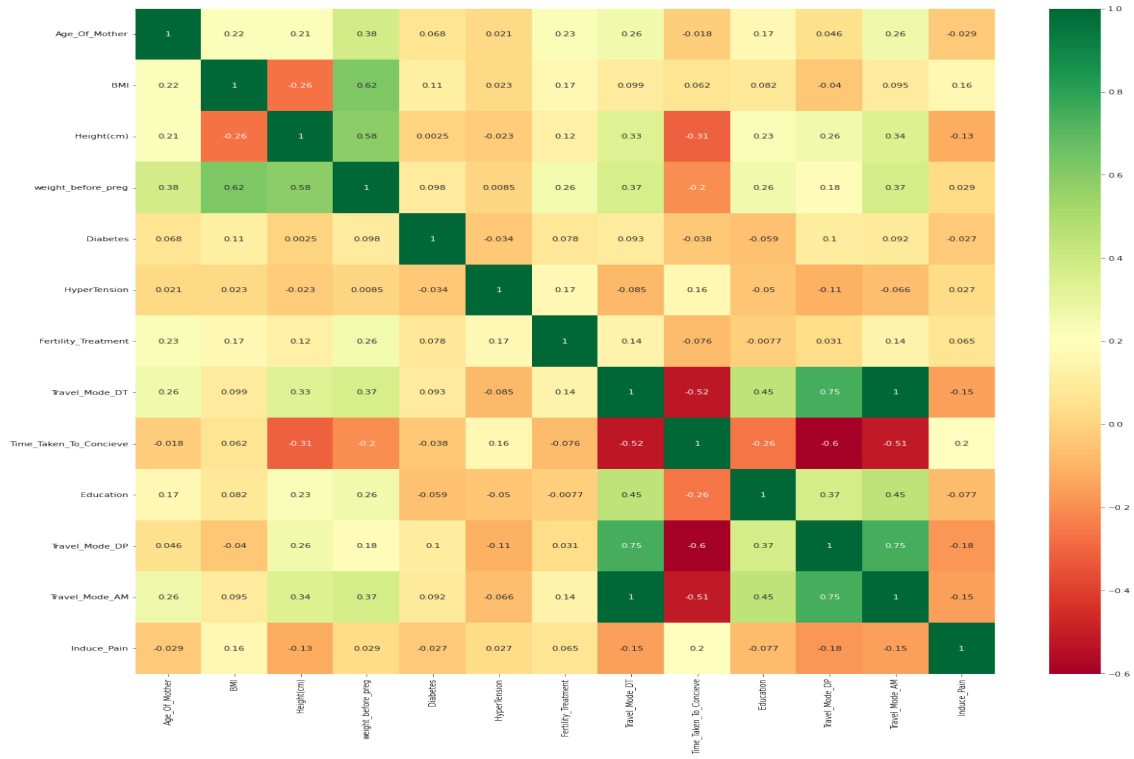

Figure 5.15: **Correlation matrix with heatmap for "Induce Pain"**

Just like the target variable "PreTerm", other target variables show similar properties. After observing the correlation heatmap of both "PreTerm" and "Full Term" in **Figure 5.8** and **Figure 5.9**, we can say that both have almost the same relation with the features but the difference is in the type of correlation they have with them as are the exact opposite of the other ones. For example, the positive correlations in the "PreTerm" heatmap like the "Diabetes" has negative correlation in "Full Term" whereas the negative correlation in the "PreTerm" heatmap like the "Education" has positive correlation in "Full Term". This makes sense as both the targets are exact opposite from each other meaning the increase in the possibility of Preterm labour in pregnant women will decrease the possibility of full term. Also, having a negative correlation with "PreTerm" means that higher level of educational qualification decreases the possibility of preterm birth thus increasing the possibility of full term. Same could be seen for the target variables "C-section" and "Vaginal Delivery" as the possibility increase in one decreases the other.

5.7 Classification Result

Various type of classifier models were used for this research which are K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Decision Tree Classifier (DTC), Logistic regression, Random Forest Classifier (RFC), Naive Bayes Classifier and Extreme Gradient Boosting (XGBoost). At first we performed some of the classification models on our dataset to predict the Target variables "PreTerm", "Full Term", "no of births(single/Twins)", "NICU Stay", "Jaundice", "C-section", "Vaginal Delivery" and “Induce Pain” one after another. Classification performance of the classification models for each target variable are provided in the following tables.

Table 5.1: **Accuracy score for predicting preterm**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	83.33%	77.78%	x
Support Vector Machine (SVM)	83.33%	52.22%	x
Decision Tree Classifier (DTC)	76.67%	x	83.39%
Logistic Regression	81.11%	x	x
Random Forest Classifier (RFC)	83.34%	x	x
Naive Bayes Classifier	25.54%	84.44%	x
Extreme Gradient Boosting (XGBoost)	81.11%	83.34%	84.16%

Table 5.2: **Accuracy score for predicting Full Term**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	83.33%	75.56%	x
Support Vector Machine (SVM)	83.33%	83.33%	x
Decision Tree Classifier (DTC)	76.67%	x	83.05%
Logistic Regression	81.11%	x	x
Random Forest Classifier (RFC)	83.34%	x	x
Naive Bayes Classifier	25.54%	84.44%	x
Extreme Gradient Boosting (XGBoost)	81.11%	81.11%	84.16%

Table 5.3: **Accuracy score for predicting no. of birth (Single/Twin)**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	98.89%	97.78%	x
Support Vector Machine (SVM)	96.67%	97.78%	x
Decision Tree Classifier (DTC)	96.67%	x	97.5%
Logistic Regression	97.77%	x	x
Random Forest Classifier (RFC)	97.77%	x	x
Naive Bayes Classifier	25.54%	98.89%	x
Extreme Gradient Boosting (XGBoost)	97.77%	97.77%	97.77%

Table 5.4: **Accuracy score for predicting NICU Stay**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	70.00%	68.89%	x
Support Vector Machine (SVM)	72.22%	73.33%	x
Decision Tree Classifier (DTC)	62.22%	x	83.89%
Logistic Regression	66.66%	x	x
Random Forest Classifier (RFC)	78.89%	x	x
Naive Bayes Classifier	31.11%	81.11%	x
Extreme Gradient Boosting (XGBoost)	67.78%	67.78%	78.05%

Table 5.5: **Accuracy score for predicting Jaundice**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	92.22%	82.22%	x
Support Vector Machine (SVM)	86.67%	87.78%	x
Decision Tree Classifier (DTC)	83.33%	x	89.45%
Logistic Regression	84.44%	x	x
Random Forest Classifier (RFC)	88.89%	x	x
Naive Bayes Classifier	38.89%	87.78%	x
Extreme Gradient Boosting (XGBoost)	84.44%	86.67%	89.16%

Table 5.6: **Accuracy score for predicting C-section**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	51.11%	51.11%	x
Support Vector Machine (SVM)	56.67%	57.78%	x
Decision Tree Classifier (DTC)	48.88%	x	53.61%
Logistic Regression	54.44%	x	x
Random Forest Classifier (RFC)	58.89%	x	x
Naive Bayes Classifier	47.78%	58.89%	x
Extreme Gradient Boosting (XGBoost)	50.00%	48.89%	56.67%

Table 5.7: **Accuracy score for predicting Vaginal Delivery**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	51.11%	40.00%	x
Support Vector Machine (SVM)	55.56%	56.67%	x
Decision Tree Classifier (DTC)	50.00%	x	53.61%
Logistic Regression	52.22%	x	x
Random Forest Classifier (RFC)	58.89%	x	x
Naive Bayes Classifier	47.78%	58.89%	x
Extreme Gradient Boosting (XGBoost)	53.33%	51.11%	57.22%

Table 5.8: **Accuracy score for predicting Induce Pain**

Name of the Classifier	Accuracy score before feature selection	Accuracy score after feature selection	Accuracy score after Fine-tuning
K-Nearest Neighbors (KNN)	96.67%	98.89%	x
Support Vector Machine (SVM)	94.44%	97.78%	x
Decision Tree Classifier (DTC)	92.23%	x	96.66%
Logistic Regression	92.23%	x	x
Random Forest Classifier (RFC)	97.77%	x	x
Naive Bayes Classifier	75.55%	92.22%	x
Extreme Gradient Boosting (XGBoost)	98.89%	98.89%	96.66%

It can be seen how much the performance has improved after the feature selection was done using all three methods especially in Naive Bayes Classifier. Moreover, fine-tuning of the Decision Tree Classifier and XGBoost resulted in raising their performance too. After observing the accuracy score for each target variable in different classification algorithms, it can be said that the performance of the algorithms vary a lot for predicting each categorical output data even though all of them belong to the same dataset sharing the same features. Thus, it is required to use various classification methods for achieving the goal of this paper and implement the appropriate method with the best accuracy provided.

5.8 Performance of models

After understanding the relationships among the features, the datas are then trained and tested to check the performance accuracy of the predicted output by generating confusion matrix and the detailed description of the result are plotted graphically using the receiver operating characteristics curve as shown below:

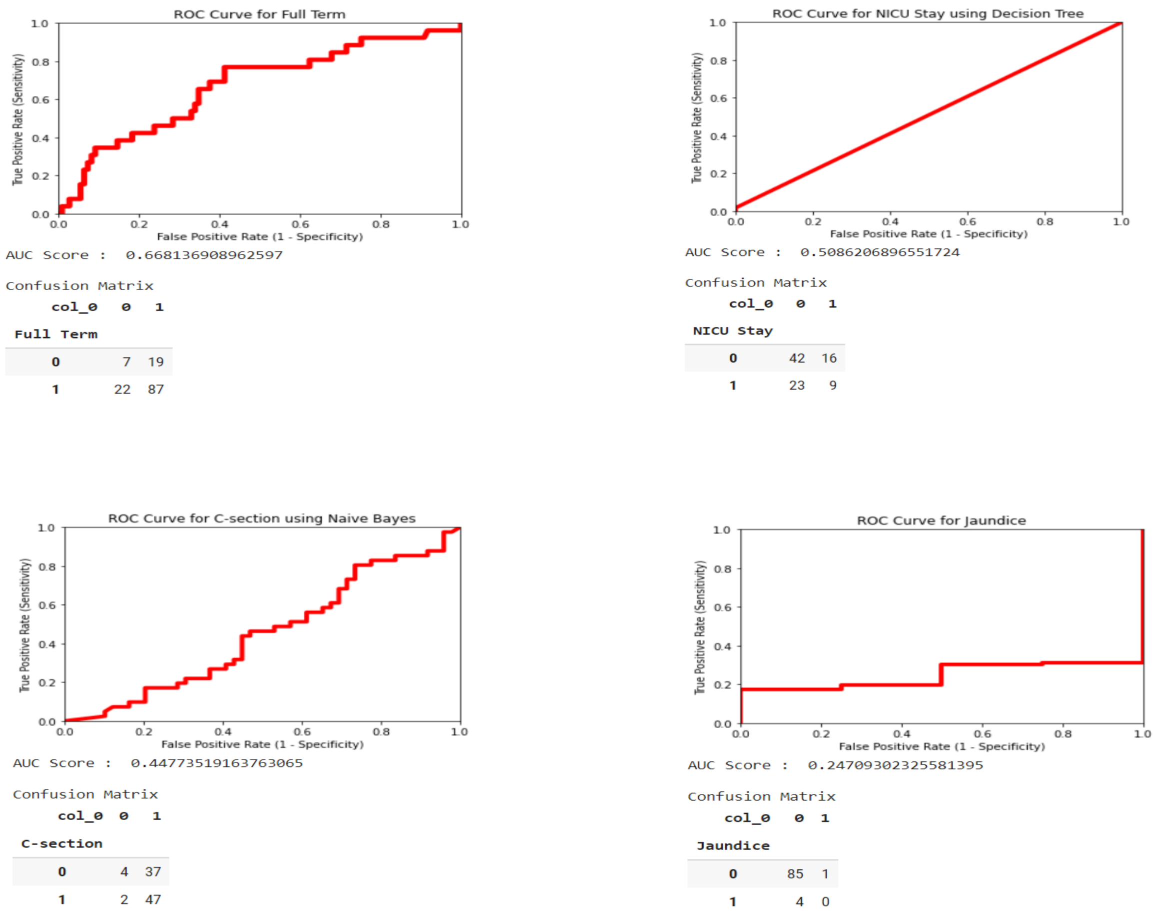

Figure 5.16: **Receiver Operating Characteristics (ROC) curve of four different predicted outputs**.

In **Figure 5.16**, four different ROC curves are shown to display the performance of the output variable with different models. The ROC curves are generated by plotting the True Positive Rate (TPR) and False Positive Rate (FPR) at various threshold settings. The true positive rate is the sensitivity, recall or probability of detection and the false positive rate is the probability of false alarm which is calculated as (1 - specificity). The true positive (TP), true negative (TN), false positive (FP), false negative (FN) values are found from the confusion matrix that is formed for each output.

The equations used to calculate these values are:

$$\begin{aligned} TPR &= (TP)/(TP + FN) \\ &= (1 - FNR) \end{aligned} \tag{5.2}$$

Where TP is true positive, FN is false negative , FNR is false negative rate.

Similarly, FNR is calculated using the formula:

$$\begin{aligned} FNR &= (FN)/(FN + TP) \\ &= (1 - TPR) \end{aligned} \tag{5.3}$$

Where FN is false negative, TP is true positive , TPR is true positive rate.

Along with the ROC curve, the AUC (area under the curve) score is shown which is the summary of the ROC curve measured to find the ability of the classifier to distinguish between classes. The score value ranges from 0 to 1. The closer the curve gets to the upper-left hand corner, the better the performance of the model and greater the AUC score.

Chapter 6

Chatbot using BERT and GPT2

6.1 Data Collection

The most challenging part of our research has been finding a suitable dataset for our topic. The scarce amount of dataset regarding pregnant women and their experience made us frantic and worried throughout our research time. Especially, since we needed a huge chunk of text data, specifically question and answer type for the architecture we have chosen to build our AI chatbot. Fortunately, we stumbled upon two forums and one online consultancy website which are TheBump.com, Babycenter.com and Practo.com respectively. The bump and baby center are online forums where pregnant women share their problems and issues so that other women like them can help or share their own experience. On the other hand, Practo.com is an online consultancy service where people can ask questions and medical professionals answer their queries and advise them. To be on the safe side, we decided to scrape data from all of these sites yet we ended up not using the forums as their questions are mostly answered by other pregnant women not medical professionals and often these answers are simply experiences but not actual answers.

Since we did not have any prior experience in web scraping, this was a fun challenge for us. We created three scrapers for each one of the websites and coded them to fetch the questions asked by pregnant women, the answers provided by doctors or others and the url of the page where it is scraped from.

In order to create negative samples, we also included tags in our dataset as context for both the question and the answers. So, we also had to integrate a keyword extraction into our scraper. We needed to find something easy and reliable. There are many keyword extraction methods out there like Rake, YAKE!, Spacy, TF-IDF etc. and we experimented with all of them. But none of them were able to give us the proper result that we were looking for. They were generating unnecessary keywords that were appearing a lot but for our keywords we were actually looking for the specific disease or problem that data mentions as it works better for our context. And, KeyBERT ended up being the solution we were looking for. This extraction technique generated keywords relating to diseases and problems that we were looking for. It was still not perfect but suitable for our work. Since, our scrapers already had multiple nested for loops and required an immense amount of time to fetch all the data, it made us learn the best advantage of keyBERT that it only requires three lines of code.

Furthermore, we also managed to find a dataset consisting of huge amount of PDF articles regarding various issues women face during pregnancy. We decided to create another dataset by manually copying and pasting the information found in these articles. We decided to use this dataset as well due to the reason that we could make a comparative study between these two datasets.

6.1.1 KeyBERT

KeyBERT is a simplistic and easy-to-use keyword extraction model that incorporates BERT embeddings to generate keywords and phrases that are most similar to a text [28]. First, it extracts embeddings from the text with BERT to create a representation for the text. Then, it extracts word embeddings for N-gram words/phrases. After that, cosine similarity is used to find the words or phrases that are closest to the context of the text. The most similar words then are identified as the keywords for that text. While there are other great papers and solutions that use similar BERT embeddings, KeyBERT does not need any training like them and can be implemented in only three lines of code.

6.2 Data Preprocessing

Since our dataset consists of web scraped data, it has texts full of special characters that are bad for any kind of NLP work. These special characters are mainly the result of emojis, flags etc. And, our models can not understand the characters so they act like noises in our data that simply have no meaning. Hence, these characters are a great threat to the performance of our models. Moreover, we basically scraped people's questions and the answers that they got online for our dataset which means our data have a lot of contractions as well. And, contractions are another threat to our work and can compromise the performance.

First of all, we converted all the text into lower case letters so that all the texts in our data will have a similar case which helps the performance for NLP problems. Different variations in input capitalization can generate different outputs or sometimes no output at all. Then, in order to remove contractions, we used python's regex library to substitute each text's all possible contractions with it's expanded form. After that, we removed all the punctuations and brackets using regex. While we were able to remove some specific special characters using regex library's sub function, there were so many others that we knew and did not know of in the 11 thousands worth of data. The solution that we could find was to encode the texts into ascii and then decode it again. This technique seemed to have removed all possible special characters.

As our data had many special characters, by removing them we created a lot of white spaces in their stead. Hence, finally we used a regular expression to remove all those white spaces.

After analyzing the different percentiles of question and answer lengths using numpy library, we were able to find that 100% of the questions and 99% of the answers were within 500 words. So, in order for all of our questions and answers to have an uniform length, we truncated all of them to the length of 500.

6.2.1 Data labeling

[11] explains in an easy approach, the suggestion of [9] that negative-sampling approach is a more efficient way of deriving word embeddings. Although, negative-sampling incorporates skip-gram model, it is actually optimizing a different objective. Let us look at the derivation of the negative-sampling objective explained in [11].

Let us take a pair (w, c) where w denotes word and c is context. How do we know if this pair came from the training data? Here, consider $p(D = 1|w, c)$ as the probability that (w, c) exists in the corpus data. Similarly, $p(D = 0|w, c) = 1p(D = 1|w, c)$ is the probability that (w, c) did not come from the corpus data. Let's assume that there are parameters θ which is responsible for the distribution $p(D = 1|w, c; \theta)$.

Now, we need to find the parameters which maximize the probabilities that all of the observations came from the data.

$$
\begin{aligned}
&arg\max_{\theta} \prod_{(w,c)\in D} p(D = 1|w, c; \theta) \\
=&arg\max_{\theta} \log \prod_{(w,c)\in D} p(D = 1|w, c; \theta) \\
=&arg\max_{\theta} \sum_{(w,c)\in D} p(D = 1|w, c; \theta)
\end{aligned}
\tag{6.1}
$$

Using softmax we can define the quantity as $p(D = 1|w, c; \theta)$ as:

$$
p(D = 1|w, c; \theta) = \frac{1}{1 + e^{-v_c \cdot v_w}} \tag{6.2}
$$

Leading to the objective,

$$
arg\max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1 + e^{-v_c \cdot v_w}} \tag{6.3}
$$

This has a minor solution if we decide to set θ such that $p(D = 1w, c;) = 1$ for every pair (w, c). This can be implemented fairly easily if we set θ such that $v_c = v_w$ and $v_c v_w = K$ for all v_c, v_w, where K is quite a large number.

Now, we need to find a technique that makes sure that all the vectors do not have the same value, by preventing some (w, c) combinations. A possible approach would be to generate some (w, c) pairs for the model which have a low value of $p(D = 1|w, c;)$, i.e. pairs that do not exist in the data. This can be accomplished by generating the set D' consisting of random $(w.c)$ pairs while marking them as all incorrect. And, this is where the name "negative-sampling" originates from which is the set $D^{'}$ of randomly sampled negative examples. The optimization objective is now as follows:

$$
\begin{aligned}
& arg \max_{\theta} \prod_{(w,c)\in D} p(D=1|c,w;\theta) \prod_{(w,c)\in D'} p(D=0|c,w;\theta) \\
=& arg \max_{\theta} \prod_{(w,c)\in D} p(D=1|c,w;\theta) \prod_{(w,c)\in D'} (1-p(D=1|c,w;\theta)) \\
=& arg \max_{\theta} \sum_{(w,c)\in D} \log p(D=1|c,w;\theta) \sum_{(w,c)\in D'} \log(1-p(D=1|c,w;\theta)) \\
=& arg \max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c\cdot v_w}} \sum_{(w,c)\in D'} \log(1-\frac{1}{1+e^{-v_c\cdot v_w}}) \\
=& arg \max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c\cdot v_w}} \sum_{(w,c)\in D'} \log(\frac{1}{1+e^{v_c\cdot v_w}})
\end{aligned} \tag{6.4}
$$

If we assume $\sigma(x) = \frac{1}{1+e^{-x}}$ we get:

$$
\begin{aligned}
& arg \max_{\theta} \sum_{(w,c)\in D} \log \frac{1}{1+e^{-v_c\cdot v_w}} \sum_{(w,c)\in D'} \log(\frac{1}{1+e^{v_c\cdot v_w}}) \\
=& arg \max_{\theta} \sum_{(w,c)\in D} \log \sigma(v_c\cdot v_w) \sum_{(w,c)\in D'} \log \sigma(-v_c\cdot v_w)
\end{aligned} \tag{6.5}
$$

Here, the first term maximizes the probability of occurrence for actual pairs that exist in our data and the second term, iterates over random negative samples D' and minimizes the probability of occurring.

Since this approach produces good word representation, we decided to use it for our model. We achieved this by first labeling all of our original data, the question and answer pairs with positive 1.0. Then, we generated some random samples Q', A', T' and using our tags as context, by randomly picking up a tuple, we check if the intersection between the original tag T and the sample T' is null. If the intersection is null, we take the answer A' for that corresponding tag T' and pair it with the original question Q, thus creating a negative sample. However, if the intersection is not null, we repeat the process and keep checking for intersection. Finally, by concatenating both the positive labeled and negative labeled datasets, our final labelled dataset was created.

6.2.2 Tokenizing, Filtering, Padding and Masking

All sentences must be padded or truncated to a single and fixed length and therefore after tokenizing our data, we used the maximum sequence length of 512 for our padding. Since, 99% of all question and answers were within 500 words, we were able to maintain the default pre-trained weights of BioBert and also we do not suffer any loss of information. We also needed to prepare attention masks for our data and pass them to our bert model since padding creates empty tokens in order for smaller sequences to match the determined length. And, attention mask helps the model focus on actual content rather than these padded tokens.

6.3 Implementation

6.3.1 Fine tuned Biobert model

This model has been created consisting of a shared pre-trained BioBERT model as well as two residual blocks of FFN with dropouts each dedicated to the questions and answers of our data.

Here, we get the embeddings for our questions and answers from BioBERT and then pass it to the residual blocks. The reason behind having two residual blocks is that the model needs to be able to directly pass the embeddings if it finds it to best. And, dropouts are to make sure it does not suffer from overfitting. When a network is too closely fit to a limited set of input samples, overfitting occurs. By using dropouts, we are letting the network to concentrate on other features, hence, avoiding risks of overfitting.

After, getting the result embeddings from the residual blocks for both question and answers, the cosine similarity between them is corrected. In the end, we train our model considering mean squared error loss between the expected cosine similarities -1.0 as well as 1.0 and the cosine similarities that we obtained from the model. We choose the best possible model depending on their performance on both train and validation data, while experimenting with different thresholds to classify between positive and negative classes.

6.3.2 Question and Answer Embedding Extraction

After training the model, questions and answers are passed to the model through its residual block in order to get the embeddings for each of them. These embeddings are then stored besides the original question and answers for semantic search using Facebook's FAISS (**Figure 6.1**).

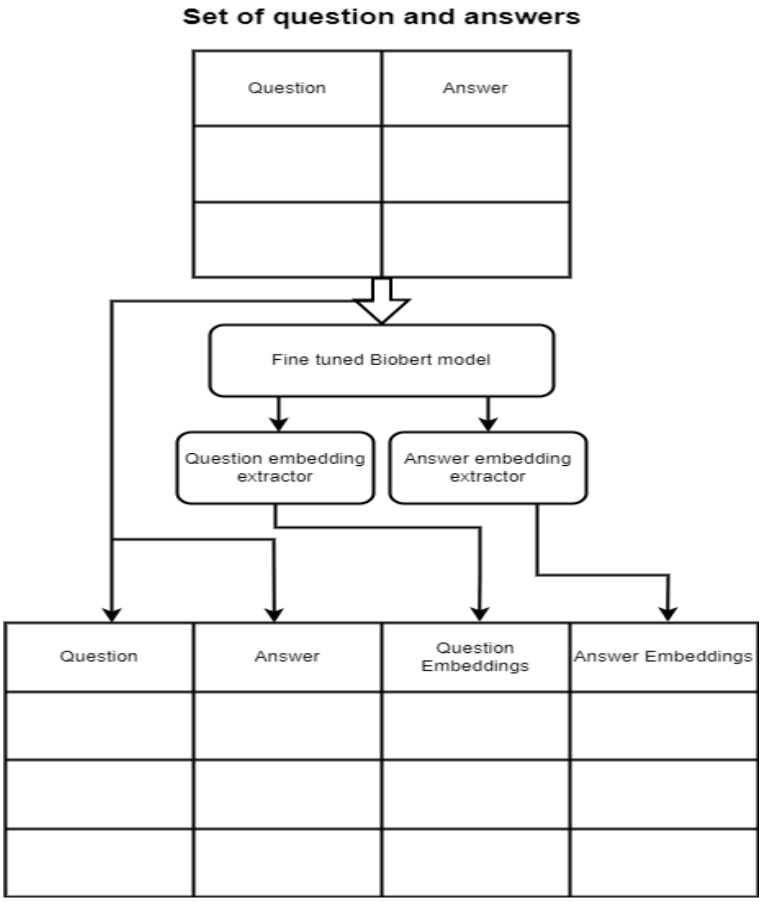

Figure 6.1: **question and answer embedding extraction**

6.3.3 FAISS and GPT-2

After training the BioBert and extracting embeddings of the question and its corresponding answers, we move on to fine-tuning the GPT-2. We use the FAISS library used in Facebook to fetch similar answers and corresponding questions by performing a semantic search on each question in the training dataset. Cosine similarity FAISS gives a score indicating how similar the pre-computed embeddings of each answer and current questions are. This can also provide similar results. After that, we concatenate the received similarly ranked question-answer pairs in order with the original question and its corresponding answer to the following string shown in **Figure 6.2** .

Explicitly written “Question” and “Answer” help the model know when they should expect a question and when to expect an answer. After fine-tuning, we proceed to train the hugging face GPT-2 model using TF gradient by feeding it the similar question and answer pairs ’Q2A2Q1A1QA’ so that it learns to take ’Q2A2Q1A1Q’ as context to generate the answer ’A’. To make sure the model follows this way of learning, loss mask was needed to be calculated . So in short, the question from a user-patient will first pass to the question extractor model to get the question embeddings, these embeddings are used to fetch similar question-answer pairs, and then the semantically related question answers are used to generate the answer to the target question using GPT2’s generative modeling.

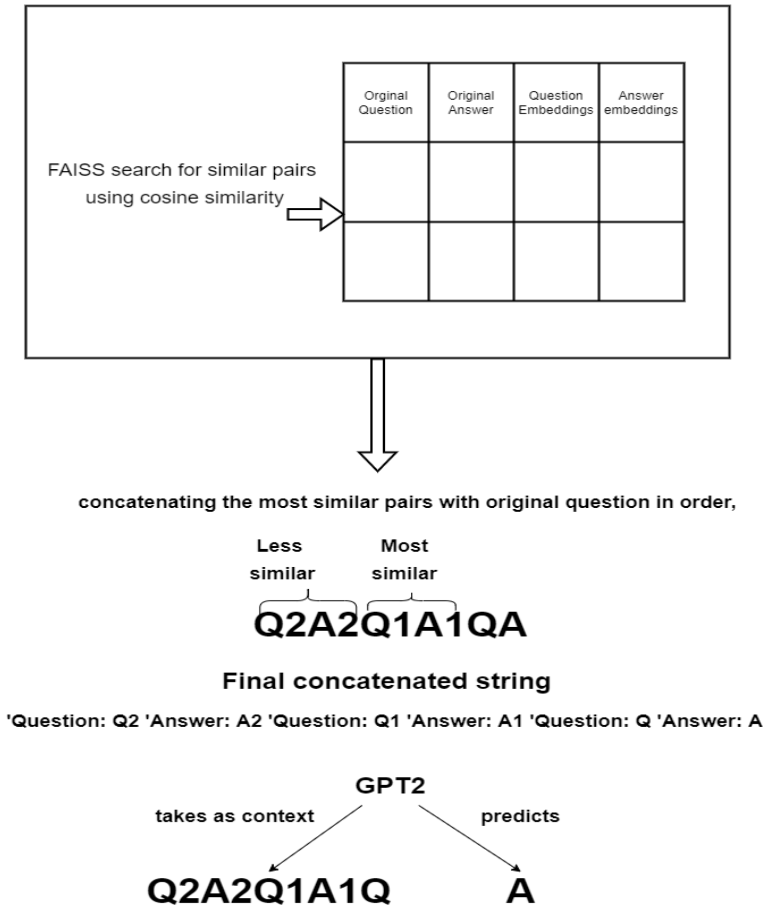

Figure 6.2: **Faiss and GPT2**

6.3.4 Model Evaluation

For Web Scraped data

The model was evaluated based on the cosine similarities it returned for the validation dataset, both for negative and positive points it manages to classify correctly while also considering the cosine similarities it returned for incorrectly classifying points that are negative and positive points originally.

In **Figures 6.3 and 6.4**, it can be seen that the correctly predicted positive and negative points are fairly separated from each other and the model obtained best accuracy for both train and validation data at the threshold 0.2. On the other hand, incorrectly classified positive and negative points are fairly close to 0.2 which shows that our model is able to separate positive and negative points when differentiating correctly and even when it fails to classify correctly, the amount is quite small.

For manually extracted data

Although, **Figures 6.5 and 6.6** show that for manually extract data, how different the plots are from the ones we showed for webscraped data. Since, our data that we

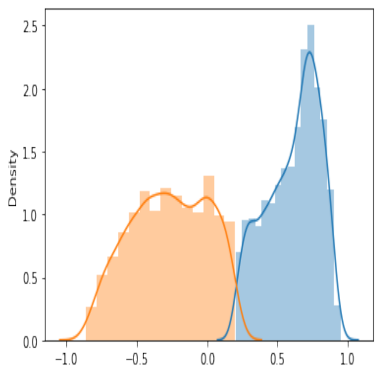

Figure 6.3: **plot for correctly predicted positive and negative points (for webscraped data)**

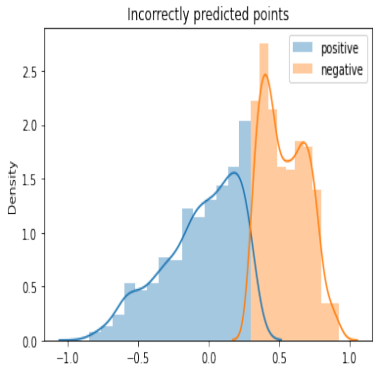

Figure 6.4: **plot for incorrectly predicted points (for webscraped data)**

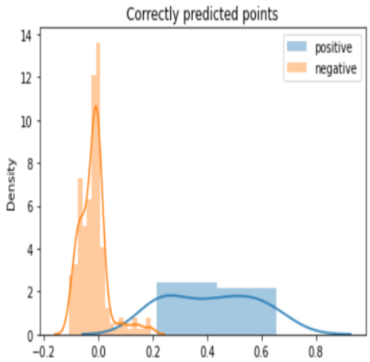

Figure 6.5: **plot for correctly predicted positive and negative points (for manually extracted data)**

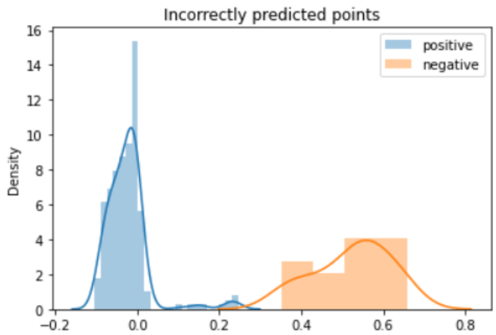

Figure 6.6: **plot for incorrectly predicted points (for manually extracted data)**

made manually by copy pasting is very small in size (1366), it is expected that we would not get good results.
Since, this trained model gave the best accuracy for both train and validation data, we chose to use this to extract our question and answer embeddings.

6.4 Results

6.4.1 For Web Scraped Data

As it was shown in the Model Evaluation section, we chose the model that obtained 79% accuracy on classifying positive and negative points correctly. So, using the model and finally implementing the chatbot, we get the results shown in **Figures 6.7 to 6.10**.

In **Figures 6.7 and 6.9**, we can see the chatbot is giving accurate answers but in **Figures 6.8 and 6.10**, the answers were not quite what we were expecting.

Figure 6.7: **result 1 (web scraped data)**

Figure 6.8: **result 2 (web scraped data)**

Although, while the answer in **Figure 6.9** was not correct, it had a similar context as it talked about baby not responding. Given that our model gave 79 percent accuracy, this was not far from our expectation.

6.4.2 For manually created data

The size of data was few in terms of NLP tasks, so it was interesting to see what we could get as results. We chose the model that gave 60 percent accuracy and after building our chat on top of that, we got the following results shown in **Figures 6.11 and 6.14**.

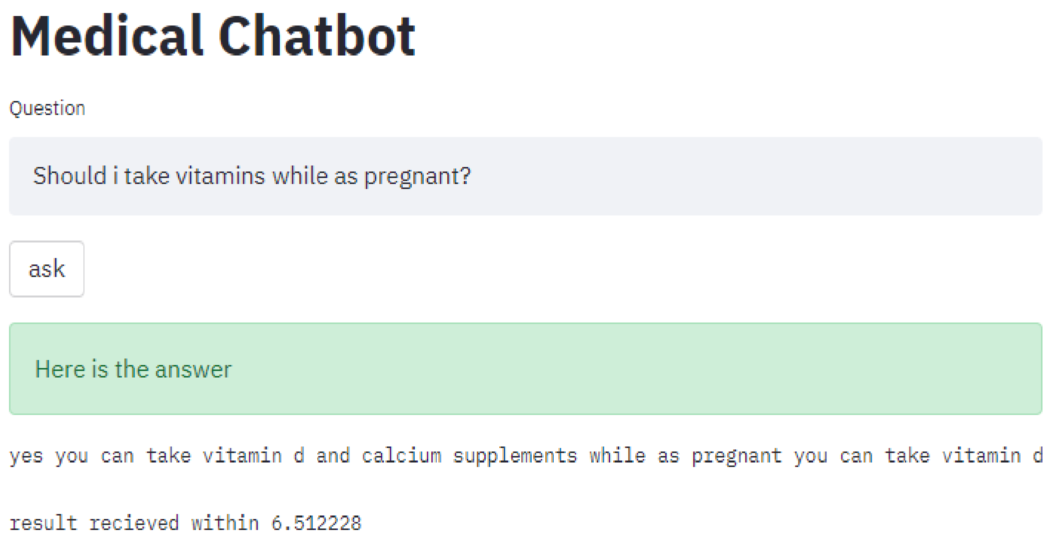

Figure 6.9: **result 3 (web scraped data)**

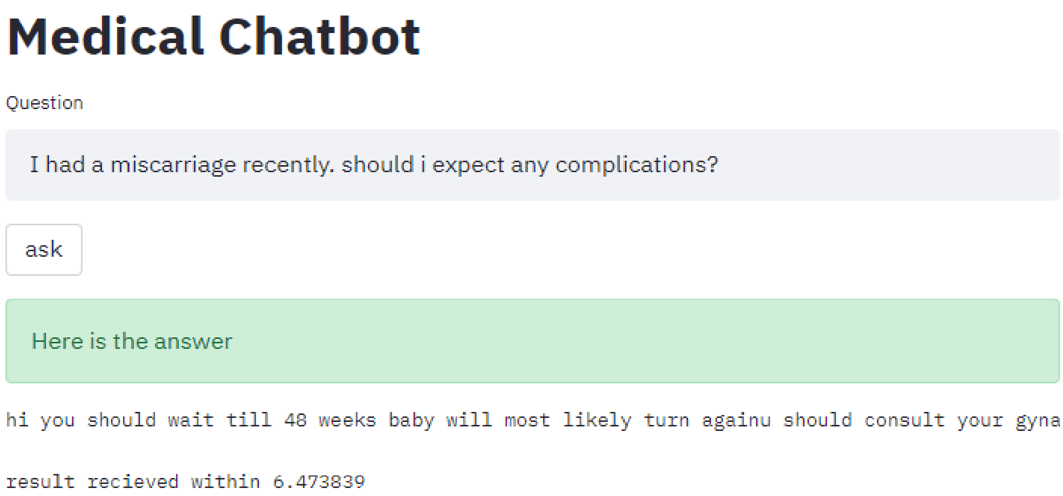

Figure 6.10: **result 4 (web scraped data)**

Figure 6.11: **result 1 (manually extracted data)**

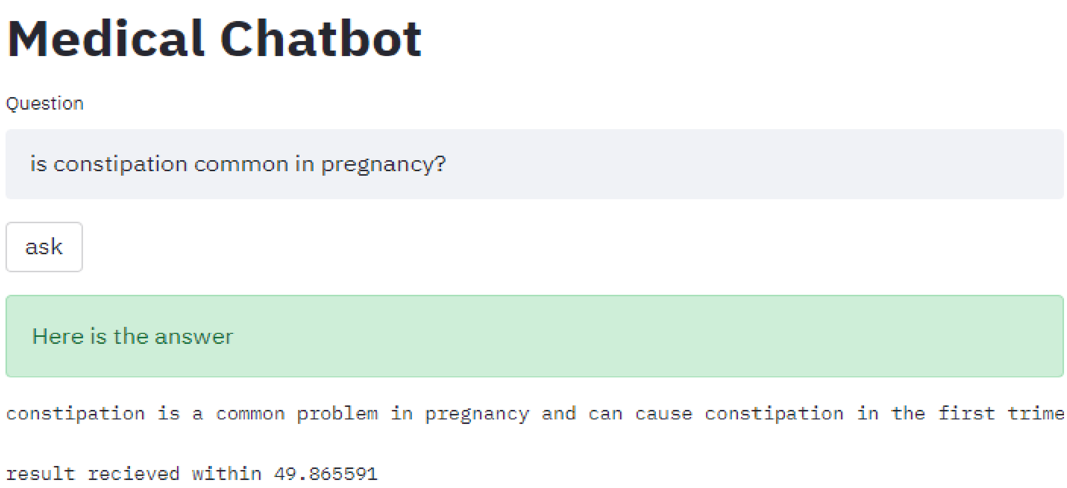

Figure 6.12: **result 2 (manually extracted data)**

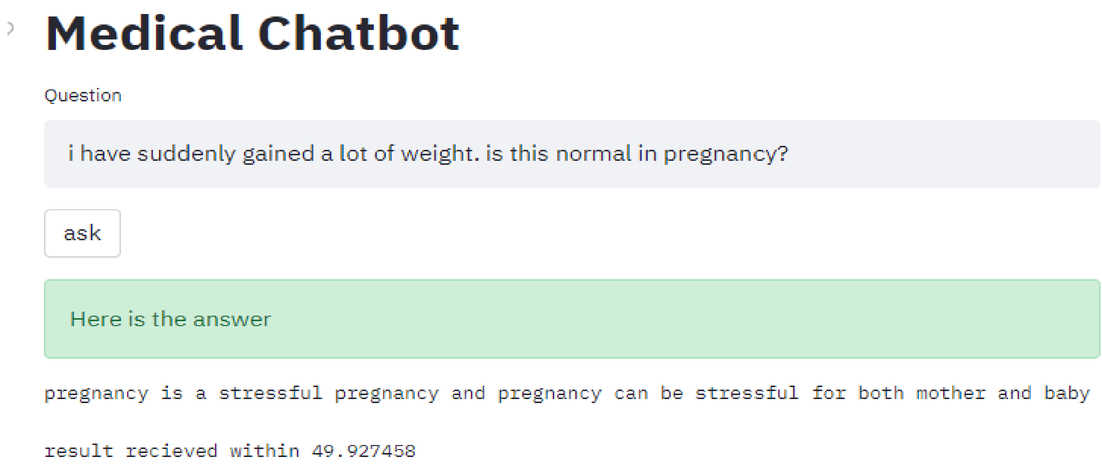

Figure 6.13: **result 3 (manually extracted data)**

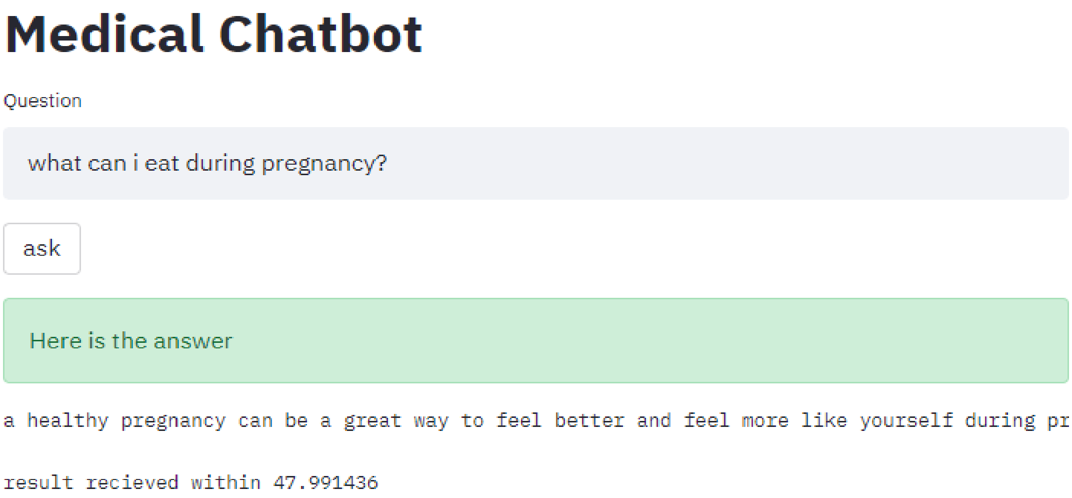

Figure 6.14: **result 4 (manually extracted data)**

Chapter 7

System Deployment

7.1 Deployment Method

In order to showcase a visual interface, a basic web application has been developed based on the factors concerning the fetal birth weight. Some of the main key features are being extracted from the 'Mother's Significant Feature (MSF)" dataset, which are then preprocessed and trained using Linear Regression Model. As the input data are continuous, Linear Regression Model has been a best choice for us with its accuracy nearest to eighty percent. The Front-End of our system has been developed using Hypertext Markup Language (HTML), cascading stylesheet (CSS), JavaScript and Bootstrap framework which is simply a two-page website. Based on some personal details and daily activities of the pregnant mother, the system takes input from the user and predict an outcome of the child's birth weight in kilogram. It displays three possible outcomes depending on inputs which are 'low birth weight', 'normal birth weight' and 'abnormal birth weight'. It is categorized depending on the output weight value of the fetus. Weight below 2.5 kg are considered to be under 'low birth weight'. In between 2.5 and 4.5, it is said to have a 'normal birth weight'. However, the child is predicted to have an abnormal birth weight if the weight is above 4.5 kg which is a rare occurrence.

After implementing the user interface, python has been used to connect the inputs in the back-end and display the outcome accordingly. For deployment, Flask, an API of python is used because of its easy functionalities and usage which make the code and learning easier. The inputs given by the user are collected and listed in an array which are then serialized using the pickle module. These python objects are then changed to binary values by this module which can also be changed back to its original form which is referred to as 'unpickle'. After all the connections have been made, the 'app.py' file, which connects both front and back-end of the system, is run on Anaconda Command prompt that provides a local Ip address to execute and run the website, displaying a predicted weight of the fetus. **Figure 7.2** shows the user interface as well as the predicted fetal birth weight outcome from randomly placed inputs.

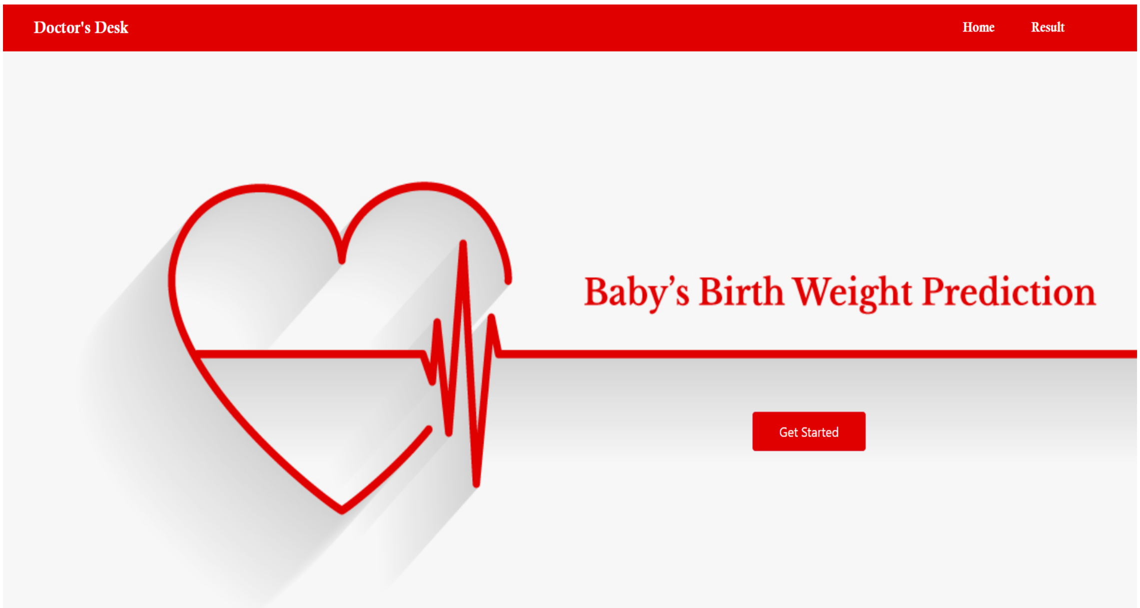

Figure 7.1: **Home Page**

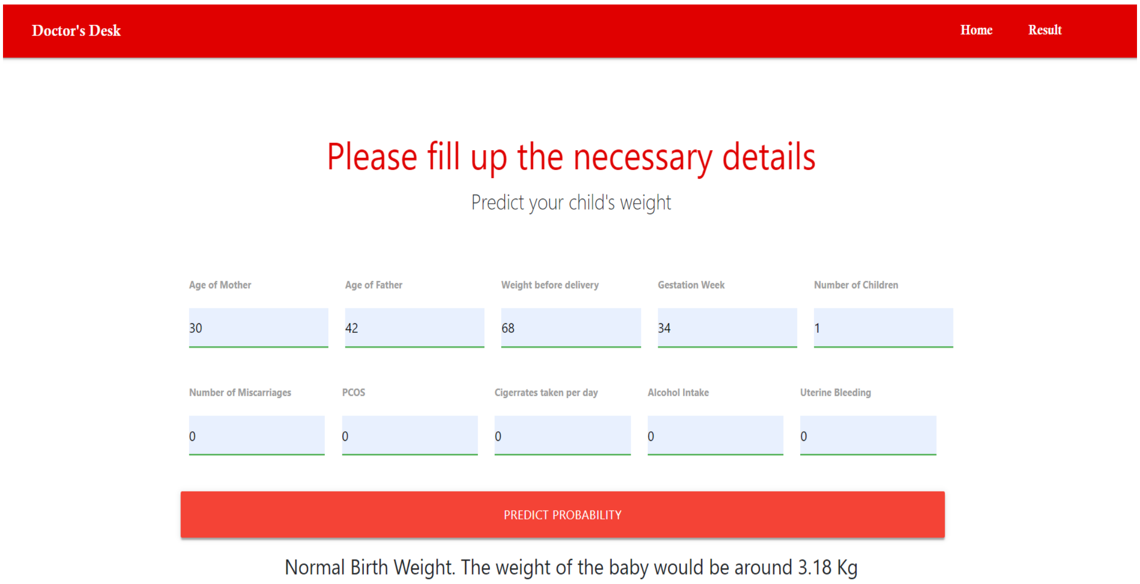

Figure 7.2: **Predicted value of the output of the fetus's weight based on input**

7.2 Limitations and Improvements

A shortage of appropriate data has been a great issue in our research. Thus, more relevant and appropriate models can help us predict more accurately with a better outcome. Besides, the medical chatbot that we have presented for the assistance of pregnant mother, can be further integrated to a modern system to provide both mental and physical assistance while presenting the possibilities of having twins, c-section, vaginal delivery or even miscarriages so that necessary precautions can be taken at the earliest convenience. We are highly expecting heading towards such modern and digital way of assisting our precious mothers in the long-run. Moreover, we aim to deploy our developed application to a SSL certified domain and hosting as well as make it a responsive user-friendly mobile and web based application so that it can be accessed by everyone from everywhere.

Therefore, we are expecting to have more contributions on our proposal of the development of a smart system that can intelligently assist people with both mental and physical support based on the previous records or data and would aid pregnant women to determine the likelihood of an epidemic, advising the user to contact doctors in these occurrences. Such contributions would, someday, be a life changing tool for us.

Chapter 8

Conclusion

8.1 Conclusion

In the challenging time of pregnancy, technological support like assisted living will serve pregnant women with proper medical treatment and improve women's experience of pregnancy and birth. Women nowadays tend to seek help from technology, especially search engines, to overcome their concerns and information needs regarding their pregnancy. However, most of the systems that came up to support pregnant women chiefly focus on their physical health. But researches have proved the necessity of incorporating mental, social and overall life support of the pregnant woman while designing mHealth apps, rather than merely reinforcing medical guidelines for health. In addition, applications should be convenient enough for women of all classes. Moreover, most of the systems use such algorithms for diagnosing that do not perform much well in case of accuracy. To overcome this problem, this research paper came up with the concept of using various kinds of classifier techniques and choosing the most suitable one to perform the diagnosis. Thus, with an aim to improve and overcome the lacking of the existing mHealth apps designed for pregnant women, this research attempts to suggest a system that will cover all aspects of pregnancy, including a chatbot to make it more convenient for women of all classes to satisfy their curiosity and queries, as well as provide results with higher accuracy by using the model that performs best among the various kinds and advanced models.

Bibliography

[1] F. Jelinek, "Interpolated estimation of markov source parameters from sparse data," in *Proc. Workshop on Pattern Recognition in Practice, 1980*, 1980.

[2] K. Kira and L. A. Rendell, "A practical approach to feature selection," in *Machine learning proceedings 1992*, Elsevier, 1992, pp. 249–256.

[3] T. Evgeniou and M. Pontil, "Support vector machines: Theory and applications," in *Advanced Course on Artificial Intelligence*, Springer, 1999, pp. 249–257.

[4] B. M. Sibai, S. N. Caritis, J. C. Hauth, *et al.*, "Preterm delivery in women with pregestational diabetes mellitus or chronic hypertension relative to women with uncomplicated pregnancies," *American journal of obstetrics and gynecology*, vol. 183, no. 6, pp. 1520–1524, 2000.

[5] I. Rish *et al.*, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, 2001, pp. 41–46.

[6] C.-Y. J. Peng, K. L. Lee, and G. M. Ingersoll, "An introduction to logistic regression analysis and reporting," *The journal of educational research*, vol. 96, no. 1, pp. 3–14, 2002.

[7] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[8] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 272, 2012.

[9] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing Systems*, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds., vol. 26, Curran Associates, Inc., 2013, pp. 3111–3119. [Online]. Available: https://proceedings.neurips.cc/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf.

[10] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[11] Y. Goldberg and O. Levy, "Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method," Feb. 2014.

[12] T. Peyton, "Pregnancy ecologies as teachable moments for the lifecourse: Changing the mhealth design paradigm," in *Proceedings of the 18th International Conference on Supporting Group Work*, 2014, pp. 269–271.

[13] T. Peyton, E. Poole, M. Reddy, J. Kraschnewski, and C. Chuang, “” every pregnancy is different” designing mhealth for the pregnancy ecology,” in *Proceedings of the 2014 conference on Designing interactive systems*, 2014, pp. 577–586.

[14] ——, “Information, sharing and support in pregnancy: Addressing needs for mhealth design,” in *Proceedings of the companion publication of the 17th ACM conference on Computer supported cooperative work & social computing*, 2014, pp. 213–216.

[15] A. Fourney, R. W. White, and E. Horvitz, “Exploring time-dependent concerns about pregnancy and childbirth from search logs,” in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 737–746.

[16] M. W. Moreira, J. J. Rodrigues, A. M. Oliveira, and K. Saleem, “Smart mobile system for pregnancy care using body sensors,” in *2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT)*, IEEE, 2016, pp. 1–4.

[17] G. Saranya, G. Geetha, and M. Safa, “E-antenatal assistance care using decision tree analytics and cluster analytics based supervised machine learning,” in *2017 International Conference on IoT and Application (ICIOT)*, IEEE, 2017, pp. 1–3.

[18] H. Sun and X. Hu, “Attribute selection for decision tree learning with class constraint,” *Chemometrics and Intelligent Laboratory Systems*, vol. 163, pp. 16–23, 2017.

[19] S. N. Tumpa, A. B. Islam, and M. T. M. Ankon, “Smart care: An intelligent assistant for pregnant mothers,” in *2017 4th International Conference on Advances in Electrical Engineering (ICAEE)*, IEEE, 2017, pp. 754–759.

[20] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[21] N. Yalina, D. R. Santi, and M. A. Aziz, “Early warning information system of pregnancy risk as an effort to reduce maternal mortality rate,” in *Proceedings of the International Conference on Algorithms, Computing and Systems*, 2017, pp. 50–54.

[22] N. Bagalkot, N. Verdezoto, M. Lewis, *et al.*, “Towards enhancing everyday pregnancy care: Reflections from community stakeholders in south india,” in *Proceedings of the 9th Indian Conference on Human Computer Interaction*, 2018, pp. 71–74.

[23] H. H. Patel and P. Prajapati, “Study and analysis of decision tree based classification algorithms,” *International Journal of Computer Sciences and Engineering*, vol. 6, no. 10, pp. 74–78, 2018.

[24] A. B. Queyam, R. K. Meena, S. K. Pahuja, and D. Singh, "An iot based multi-parameter data acquisition system for efficient bio-telemonitoring of pregnant women at home," in *2018 8th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*, IEEE, 2018, pp. 14–15.

[25] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, *Bert: Pre-training of deep bidirectional transformers for language understanding*, 2019. arXiv: 1810.04805 [cs.CL].

[26] W. F. Mustika, H. Murfi, and Y. Widyaningsih, "Analysis accuracy of xgboost model for multiclass classification-a case study of applicant level risk prediction for life insurance," in *2019 5th International Conference on Science in Information Technology (ICSITech)*, IEEE, 2019, pp. 71–77.

[27] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[28] M. Grootendorst, *Keybert: Minimal keyword extraction with bert.* Version v0.3.0, 2020. DOI: 10.5281/zenodo.4461265. [Online]. Available: https://doi.org/10.5281/zenodo.4461265.

[29] J. Lee, W. Yoon, S. Kim, *et al.*, "Biobert: A pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[30] N. Verdezoto, F. Carpio-Arias, V. Carpio-Arias, *et al.*, "Indigenous women managing pregnancy complications in rural ecuador: Barriers and opportunities to enhance antenatal care," in *Proceedings of the 11th Nordic Conference on Human-Computer Interaction: Shaping Experiences, Shaping Society*, 2020, pp. 1–9.

[31] P. Cunningham and S. J. Delany, "K-nearest neighbour classifiers-a tutorial," *ACM Computing Surveys (CSUR)*, vol. 54, no. 6, pp. 1–25, 2021.

[32] H. Deshpande and L. Ragha, "Mother's lifestyle feature relevance for nicu and preterm birth prediction," in *ITM Web of Conferences*, EDP Sciences, vol. 40, 2021, p. 03 039.

[33] S. Jammalamadaka, *Medical chatbot using bert and gpt2*, Feb. 2021. [Online]. Available: https://suniljammalamadaka.medium.com/medical-chatbot-using-bert-and-gpt2-62f0c973162f.