DEMYSTIFYING PREGNANCY HEALTH PREDICTION AND MONITORING

A PROJECT REPORT

Submitted by

ANANTHU SURESH

SNG21MCA-2008

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master Of Computer Applications

Department of Computer Applications

Sree Narayana Gurukulam College of Engineering,

Kadayiruppu, 623811

November 2022

DEMYSTIFYING PREGNANCY HEALTH PREDICTION AND MONITORING

PROJECT REPORT

Submitted by

ANANTHU SURESH (SNG21MCA-2008)

Under the Guidance of

Dr. Anoopkumar M

to

the APJ Abdul Kalam Technological University

in partial fulfilment of the requirements for the award of the Degree

of

Master Of Computer Applications

Department of Computer Applications

Sree Narayana Gurukulam College of Engineering,

Kadayiruppu, 623811

November 2022

DECLARATION

I undersigned hereby declare that the project report "**Demystifying Pregnancy Health prediction and Monitoring**", submitted for partial fulfilment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala is a bona fide work done by me under the supervision of **Dr.Anoopkumar M**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that we have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place: Kadayiruppu   **ANANTHU SURESH : SNG21MCA-2008**

Date: 25/11/2022

SREE NARAYANA GURUKULAM COLLEGE OF ENGINEERING, KADAYIRUPPU, KOLENCHERY-623811

DEPARTMENT OF COMPUTER APPLICATIONS

CERTIFICATE

This is to certify that the project report entitled "**Demystifying Pregnancy Health prediction and Monitoring**" submitted by **ANANTHU SURESH** (SNG21MCA-2008) to the APJ Abdul Kalam Technological University in partial fulfilment of the requirements for the award of the degree of Master of Computer Applications is a bona fide record of the project work carried out by him under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose**.**

Dr Anoopkumar M   **Dr. Sandhya R**

Associate Professor and Project Guide   Head of Department

Department of Computer Applications   Department of Computer Applications

Submitted to presentation and evaluation on …………….. at ……………..

Examiner 1   Examiner 2

College Seal

ACKNOWLEDGEMENT

I thank GOD almighty for guiding me throughout the project. I would like to thank all those who have supported to the completion of the project and helped me with valuable suggestions for improvement.

I owe my deep sense of gratitude to the Management of Sree Narayana Gurukulam College of Engineering, providing the best suited academic environments for the fulfilment of my project. I would like to place on record my sincere thanks to **Prof. Dr. Kemthose P Paul** Principal of our institution, for his valuable comments and suggestions to this project.

I wish to express my sincere thanks to our Head of the Department **Prof. Dr. Sandhya R**, for providing me the opportunity to undertake this project. I'm deeply indebted to my project guide **Assoc. Prof. Dr. Anoopkumar M**, and project coordinator **Asst. Prof . Vincy Devi V.K** in the Department of Computer Applications for providing me with valuable advice and guidance during the course of the project. I thank all faculty members of our department and friends for extending their cooperation during our project.

Finally, I convey my thanks to each and every one, who helped me directly and indirectly to carry out this project successfully

ANANTHU SURESH

ABSTRACT

The journey of motherhood is a life-time experience for any women, where they feel plethora of emotions of love, care and at times worry for the safety of their baby. And this very much starts as soon as one gets to find get the big news. The sudden surge of so many feelings for the unborn baby and so many unanswered questions. And without even realizing the thought of keeping the baby safe inside the womb for nine month starts making the people so jittery. However, today with the progression of technology and machine learning our app make it easy for the users to easily access all pregnancy-related information like daily health tips, weekly baby updates, GDM prediction, Risk analysis, prediction of Anemia and a lot more. There is a lot of personalized information offered in these apps as well, such as days and weeks of pregnancy, baby's development, due date of baby's arrival, personalized advice, etc. Hence, these apps serve as a dedicated companion during the entire pregnancy and meet every requirement all along.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

DEMYSTIFYING PREGNANCY HEALTH PREDICTION AND MONITORING

1 INTRODUCTION

1.1 Project Background

Whether you're a first-time momma or about to have your third baby, pregnancy will always be a wild experience. Not only is there so much going on with your own body, but you're growing another human inside you that is changing daily. Thankfully, Our app will keep you informed about what is going on each week and make sense of it all.

It can also help you monitor your body's pregnancy changes. Pregnancy is a time when your body is changing a lot, so an app can be a great way to prepare yourself for what to expect and know what's normal or not.

Our app will provide easily access to all pregnancy-related information like daily health tips, weekly baby updates, GDM prediction, Risk analysis, prediction of Anemia and a lot more. There is a lot of personalized information offered in these apps as well, such as days and weeks of pregnancy, baby's development, due date of baby's arrival, personalized advice, etc.

1.2 Existing System

- There are few apps available in the market but none of them are designed and developed according to the need of a mother.
- It is not feasible for a mother to interact with the doctor every time with each and every minute details and help.
- The post-pregnancy phase is also as essential as the pregnancy phase and needs special assistance for the mother and her child's overall development.
- The parents in rural areas usually are not aware of the basic information regarding pregnancy and parenting phase.

1.3 Proposed System

- The app will predict overall Risk level of pregnancy and can provide basic pre-caution and suggestions
- Detection of (Gestational diabetes mellitus)
- Predict whether the user have Anemia or not
- The app will notify about the daily development of the child. According to the expected delivery date, it will generate the whole timeline

2 SYSTEM ANALYSIS

2.1 Identification of Need

System analysis is the reduction of the entire system by studying various operations and their relationships with the system and the requirements of bit successor. A system can be defined as an orderly grouping of interdependent components linked together according to plan to achieve a specific objective. The idea of the system has become most practical and necessary in conceptualizing the interrelationships and integration of operations especially when using computers. Organizing consists of several interrelated and interacting components. Analysis is the detailed study of various operations performed by the system and their relations within and outside the system. During analysis, data are connected on the available files, decision points and is handled by the present system.

2.2 Preliminary Investigation

Preliminary analysis is the initial process at the start of the project that determines whether the concept is viable. It looks at economic, market, industry and social trends that influence the success of business endeavours associated with a proposed strategy. Preliminary analysis is repeated in situations where primary investigations trigger updates to plan. Conducting a preliminary analysis of a business strategy allows the organization to see the viability of an intended goal. It creates a comprehensive idea of the enterprise objective and states the outcome is meant to be expressed.

2.3 Feasibility Study

Feasibility analysis is the procedure for identifying the candidate system, evaluating and effecting the most feasible system. This is done by investigating the existing system in the area under investigation or generally ideas about a new system. It is a test of a system proposal according to its work ability, impact on the organization, ability to meet user needs, and effective use of resources feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study lets the developer for see the future of the project and the usefulness. A feasibility study of system proposal is according to its work ability, which is the impact on the organization, ability to meet their user needs and effective use of resources.

Thus, when a new application is proposed it normally goes through a feasibility study before it is approved for development.

The document provides the feasibility of the project that is being designed and lists various areas that were considered very carefully during the feasibility study of this project such as Technical, Economic, Operational, Schedule and Social feasibility's.

2.3.1 Technical Feasibility

This app will work with all android devices and web, it is built using language called dart and uses a firmware flutter. Backend of this app is firebase which is developed by google. Datasets where collected from Kaggle

So, all resources were met and this app is fully technically feasible

2.3.2 Economical Feasibility

The development of the project need less cost compare to the existing system. Hardware, design and also the overall development cost is less and the benefits are high.

2.3.3 Operational Feasibility

The project gives the services to the customer such as the maintenance, upgradation, etc.

2.3.4 Social Feasibility

This app is built using immersive care and researchs.it has most pleasing ui and ux. Every feature in this app is easily operated and understandable, Every users can easily interact with the app.

2.3.5 Schedule Feasibility

This system can be completed in the given time by dividing the project into different parts. And then solving each modules and combining it.

2.4 Project Planning

The project will go through the following stages of development in its Software Development Life Cycle.
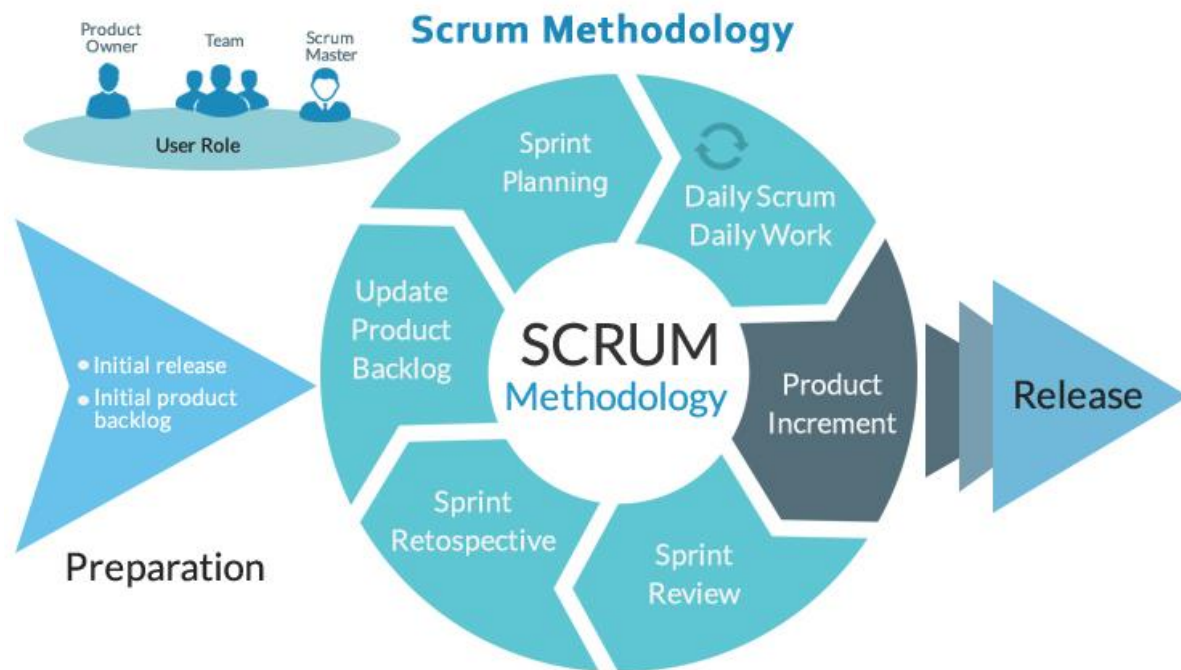
Figure 2.1: Project Planning

2.5 Development Environment

2.5.1 Flutter

Flutter is an open source framework to create high quality, high performance mobile applications across mobile operating systems - Android and iOS. It provides a simple, powerful, efficient and easy to understand SDK to write mobile application in Google's own language, Dart. This tutorial walks through the basics of Flutter framework, installation of Flutter SDK, setting up Android Studio to develop Flutter based application, architecture of Flutter framework and developing all type of mobile applications using Flutter framework.

2.5.2 Google Colab

Google Colab was developed by Google to provide free access to GPU's and TPU's to anyone who needs them to build a machine learning or deep learning model. Google Colab can be defined as an improved version of Jupyter Notebook.

- What is Jupyter Notebook?
  Jupyter Notebook is an application that allows editing and running Notebook documents through a web browser or an Integrated Development Environment (IDE). Instead of files, you will work with Notebooks.

- What is a Notebook?
  Programming Languages are an intermediate form between human-understandable language and machine understandable language. Every application is built using one of the many programming languages available. Maybe a person with a computer science background can understand, but not everyone can. Remember, as Software Developers, we develop applications for people with little computer science knowledge.
  Google Colab

   Features Google Colab provides tons of exciting features that any modern IDE offers, and much more. Some of the most exciting features are listed below.

  • Interactive tutorials to learn machine learning and neural networks.

  • Write and execute Python 3 code without having a local setup.

  • Execute terminal commands from the Notebook.

  • Import datasets from external sources such as Kaggle.

  • Save your Notebooks to Google Drive.

  • Import Notebooks from Google Drive.

  • Free cloud service, GPUs and TPUs.

  • Integrate with PyTorch, Tensor Flow, Open CV.

  • Import or publish directly from/to GitHub.

2.5.3 Visual Studio

An integrated development environment (IDE) is a feature-rich program that supports many aspects of software development. The Visual Studio IDE is a creative launching pad that you can use to edit, debug, and build code, and then publish an app.Visual Studio IDE is a comprehensive development platform for cloud, web and multiple operating systems.

It provides users with a smooth interface that is easy to navigate, allowing for faster and more accurate coding. Developers will have access to a range of debugging tools which facilitates in diagnosing and profiling bugs. This would give programmers the confidence of deploying their applications because they know that they've got a set of tools which monitors any troubles that might cause performance errors. Furthermore, Visual Studio IDE functions as a testing platform too. The IDE can be used to simulate how applications will run in their selected environments. This is for ensuring that once the application is deployed, it will run smoothly as tested.

The main benefits of Visual Studio IDE are its features that provide coding assistance to developers, rigorous testing that can be done inside the platform, support for team collaboration, and a variety of customization options. Here are more details:

- **Coding assistance**

   Regardless of the programming language being used, Visual Studio IDE offers users with a real-time coding assistant. Its built-in IntelliSense provides descriptions and hints about the API. It also helps in expediting the process through its auto-complete feature. Moreover, Visual Studio IDE makes sure that developers don't lose the last modification they made as they go through the rest of the code.

- **Testing platform**

   The IDE has a host of tools, which are available to all languages, for easily finding and diagnosing bugs. The debugging process can also be done remotely, locally, or even in the middle of production. Aside from that, Visual Studio IDE is equipped with an application testing platform which empowers developers in making sure that the products they will deploy are of high quality. Thanks to this, programmers can shift their energies on other aspects of development instead of spending too much time debugging errors.

- **Collaboration support**

   Visual Studio IDE's makers believe that more set of eyes can do better than just one which is why the platform is equipped with collaborative capabilities, thus increasing the productivity of the team. All these tools are natively integrated within the development lifecycle.

- **High customizability**

  Every user is provided with various customization options by the IDE in different forms, one of which is extending functionalities by using add-ons and extensions are available from Visual Studio Marketplace. On top of that, developers can also publish their very own extensions or add-ons.

2.5.4 Firebase

Firebase is a set of hosting services for any type of application (Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). It offers NoSQL and real-time hosting of databases, content, social authentication (Google, Facebook, Twitter and Github), and notifications, or services, such as a real-time communication server.

2.5.5 Sublime Text Editor

Sublime Text Editor is a full featured Text editor for editing local files or a code base. It includes various features for editing code base which helps developers to keep track of changes. Various features that are supported by Sublime are as follows

• Syntax Highlight

• Auto Indentation

• File Type Recognition

• Sidebar with files of mentioned directory

• Macros

• Plug-in and Packages

2.5.6 Heroku

Heroku is a cloud platform as a service (PaaS) supporting several programming languages. One of the first cloud platforms, Heroku has been in development since June 2007, when it supported only the Ruby programming language, but now supports Java, Node.js, Scala, Clojure, Python, PHP, and Go. For this reason, Heroku is said to be a polyglot platform as it has features for a developer to build, run and scale applications in a similar manner across most languages. Heroku was acquired by Salesforce in 2010 for $212 million.

2.6 Requirement Specification

2.6.1 Software Requirement

The software requirements specification (SRS) is a means of translating the ideas in the minds of clients into a formal documentation. This document forms the development and software validation. The basic reason for the difficulty in software requirement specification comes from the fact that there are three interested parties–the client, the end users and the software developer. The requirements document has to be such that the client and the user can understand easily and the developers can use it as a basis for software development .Due to the diverse parties involved in software requirement specification, a communication gap exists. This gap arises when the client does not understand software or the software development processor when the developer does not understand the client's problem and application area. SRS bridges this communication gap.

2.6.2 Hardware Requirement

Requirements analysis is the process of determining user expectations for a new or modified product. These features, called requirements, must be quantifiable, relevant and detailed. This step acquiring all the facts problem specification such as identifying the desired result determining what information is needed to produce these results and figuring out what process must be carried out to proceed to get the accurate result.

2.7 Software and Hardware Specification

2.7.1 Software Specification for development

Operating system	: Windows
Technologies	: Flutter,colab,firebase,heroku
Tools	: Vs code,Andriod Studio

2.7.2 Hardware Specification for development

Display	: 13 inch or more
RAM	: Minimum 8GB
Hark disk	: 500GB
Processor	: Intel i5 or more

2.7.3   Hardware Specification for deployment

Android device   : Minimum 30 SDK

3 SYSTEM MODELLING

3.1 Design Methodology

Design methodology refers to the development of a system or method for a unique situation. Today the term is most often applied to technological fields in reference to web design, software or information system design. The key design methodology is finding the best solution for each design situation, whether it be in industrial design, architecture or technology. Design methodology stresses the use of brainstorming to encourage innovative ideas and collaborative thinking to work through each proposed idea and arrive at the best solution. Meeting the needs and wants of the end user is the most critical concern. Design methodology also employs basic research methods, such as analysis and testing.

3.2 Use case diagram

A UML use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behaviour (what), and not the exact method of making it happen (how). Use cases once specified can be denoted both textual and visual representation (i.e. use case diagram). A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behaviour in the user's terms by specifying all externally visible system behaviour.
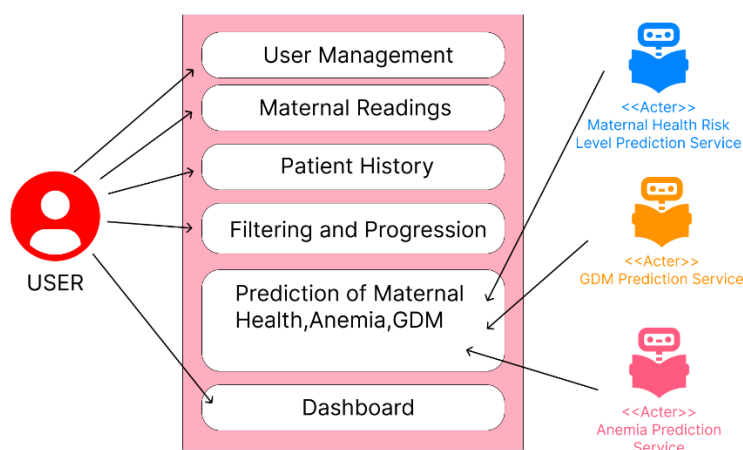
USE CASE DIAGRAM

Figure 3.1: Use case Diagram

3.3 ACTIVITY DIAGRAM

USER

PREDICTION SYSTEMS

USER

Observed Data

Decision

Maternal Health Risk Prediction

GDM Prediction
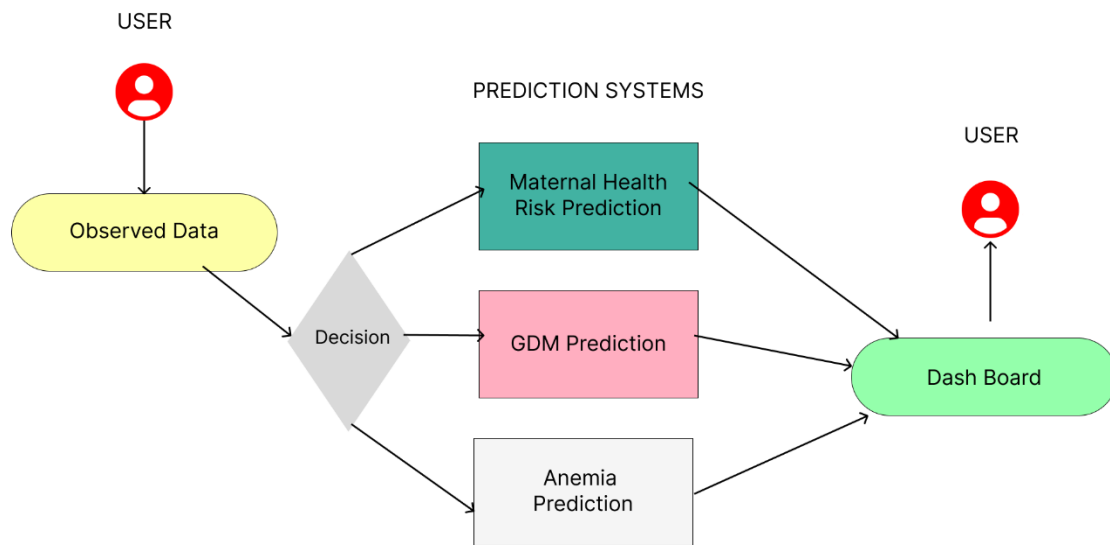
Anemia Prediction

Dash Board

Figure 3.2: Activity Diagram

4 IMPLEMENTION

4.1 how does prediction works?

It's a process that operates among multiple decision trees to get the optimum result by choosing the majority among them as the best value.

Approach:

Gathering the Data: Data preparation is the primary step for any machine learning problem. We will be using a dataset from Kaggle for this problem.

Cleaning the Data: Cleaning is the most important step in a machine learning project. The quality of our data determines the quality of our machine learning model. So it is always necessary to clean the data before feeding it to the model for training. In our dataset all the columns are numerical, the target column i.e. prognosis is a string type and is encoded to numerical form using a label encoder.

Model Building: After gathering and cleaning the data, the data is ready and can be used to train a machine learning model. We will be using this cleaned data to train the Support Vector Classifier, Naive Bayes Classifier, and Random Forest Classifier. We will be using a confusion matrix to determine the quality of the models.

Inference: After training the three models we will be predicting the disease for the input symptoms by combining the predictions of all three models. This makes our overall prediction more robust and accurate.

At last, we will be defining a function that takes symptoms separated by commas as input, predicts the disease based on the symptoms by using the trained models, and returns the predictions in a JSON format.

4.2 Prerequisites

This post works with assumptions that you are:

- Familiar with Python 3

- Good understanding of Machine learning libraries

- Have a basic understanding of Random forest Classifier

4.3 Random Forest Classifier

How the Algorithm Works and Why it Is So Effective?

A big part of machine learning is classification — we want to know what class (a.k.a. group) an observation belongs to. The ability to precisely classify observations is extremely valuable for various business applications like predicting whether a particular user will buy a product or forecasting whether a given loan will default or not.

Data science provides a plethora of classification algorithms such as logistic regression, support vector machine, naive Bayes classifier, and decision trees. But near the top of the classifier hierarchy is the random forest classifier (there is also the random forest regressor but that is a topic for another day).

we will examine how basic decision trees work, how individual decisions trees are combined to make a random forest, and ultimately discover why random forests are so good at what they do.

Decision Trees

Let's quickly go over decision trees as they are the building blocks of the random forest model. Fortunately, they are pretty intuitive. I'd be willing to bet that most people have used a decision tree, knowingly or not, at some point in their lives.
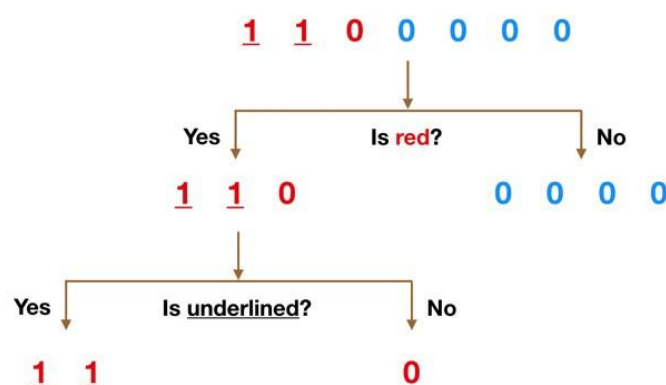
Figure 4.1: Simple Decision Tree Example

It's probably much easier to understand how a decision tree works through an example.

Imagine that our dataset consists of the numbers at the top of the figure to the left.

We have two 1s and five 0s (1s and 0s are our classes) and desire to separate the classes using their features. The features are color (red vs. blue) and whether the observation is underlined or not. So how can we do this?

Colour seems like a pretty obvious feature to split by as all but one of the 0s are blue. So we can use the question, "Is it red?" to split our first node. You can think of a node in a tree as the point where the path splits into two — observations that meet the criteria go down the Yes branch and ones that don't go down the No branch.

The No branch (the blues) is all 0s now so we are done there, but our Yes branch can still be split further. Now we can use the second feature and ask, "Is it underlined?" to make a second split.

The two 1s that are underlined go down the Yes subbranch and the 0 that is not underlined goes down the right subbranch and we are all done. Our decision tree was able to use the two features to split up the data perfectly. Victory!

Obviously in real life our data will not be this clean but the logic that a decision tree employs remains the same. At each node, it will ask

What feature will allow me to split the observations at hand in a way that the resulting groups are as different from each other as possible (and the members of each resulting subgroup are as similar to each other as possible)?

The Random Forest Classifier

Random forest, like its name implies, consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model's prediction (see figure below).

Visualization of a Random Forest Model Making a Prediction

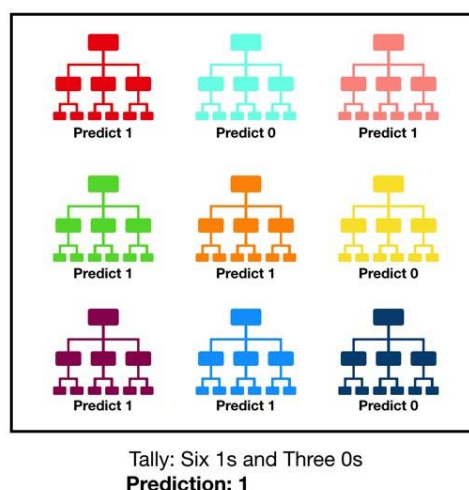Tally: Six 1s and Three 0s
Prediction: 1

Figure 4.2: Example Prediction

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

There needs to be some actual signal in our features so that models built using those features do better than random guessing.

The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

An Example of Why Uncorrelated Outcomes are So Great

The wonderful effects of having many uncorrelated models is such a critical concept that I want to show you an example to help it really sink in. Imagine that we are playing the following game:

I use a uniformly distributed random number generator to produce a number.

If the number I generate is greater than or equal to 40, you win (so you have a 60% chance of victory) and I pay you some money. If it is below 40, I win and you pay me the same amount.

Now I offer you the the following choices. We can either:

Game 1 — play 100 times, betting $1 each time.

Game 2— play 10 times, betting $10 each time.

Game 3— play one time, betting $100.

Which would you pick? The expected value of each game is the same:

Expected Value Game 1 = (0.60*1 + 0.40*-1)*100 = 20

Expected Value Game 2= (0.60*10 + 0.40*-10)*10 = 20

Expected Value Game 3= 0.60*100 + 0.40*-100 = 20

Outcome Distribution of 10,000 Simulations for each Game
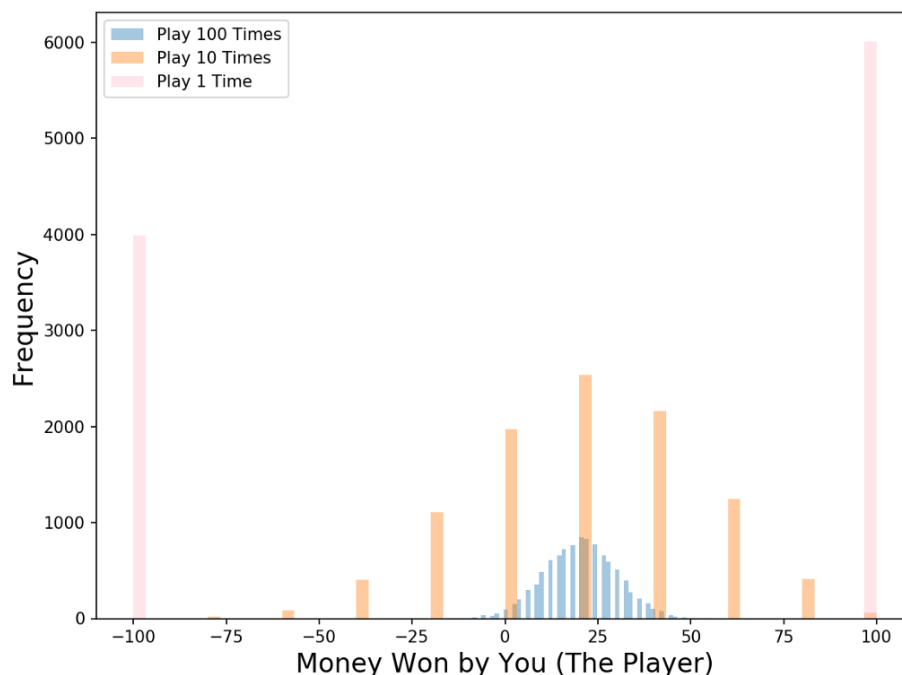
Figure 4.3: Example

What about the distributions? Let's visualize the results with a Monte Carlo simulation (we will run 10,000 simulations of each game type; for example, we will simulate 10,000 times the 100 plays of Game 1). Take a look at the chart on the left — now which game would you pick? Even though the expected values are the same, the outcome distributions are vastly different going from positive and narrow (blue) to binary (pink).

Game 1 (where we play 100 times) offers up the best chance of making some money — out of the 10,000 simulations that I ran, you make money in 97% of them! For Game 2 (where we play 10 times) you make money in 63% of the simulations, a drastic decline (and a drastic increase in your probability of losing money). And Game 3 that we only play once, you make money in 60% of the simulations, as expected.

Figure 4.4: Example

Probability of Making Money for Each Game

So even though the games share the same expected value, their outcome distributions are completely different. The more we split up our $100 bet into different plays, the more confident we can be that we will make money. As mentioned previously, this works because each play is independent of the other ones.

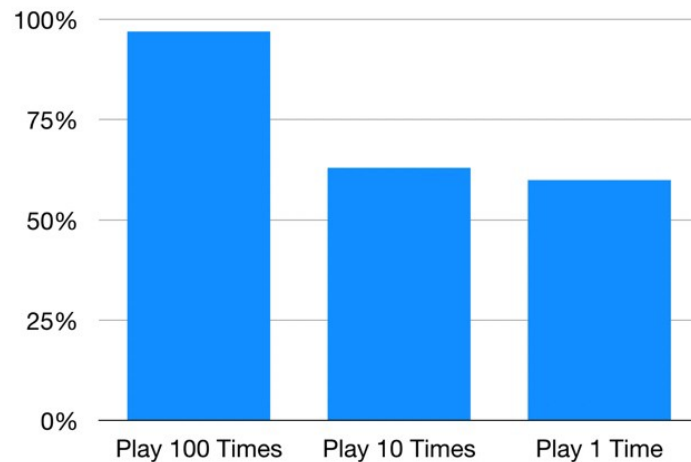Random forest is the same — each tree is like one play in our game earlier. We just saw how our chances of making money increased the more times we played. Similarly, with a random forest model, our chances of making correct predictions increase with the number of uncorrelated trees in our model.

Ensuring that the Models Diversify Each Other

So how does random forest ensure that the behavior of each individual tree is not too correlated with the behavior of any of the other trees in the model? It uses the following two methods:

Bagging (Bootstrap Aggregation) — Decisions trees are very sensitive to the data they are trained on — small changes to the training set can result in significantly different tree structures. Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset with replacement, resulting in different trees. This process is known as bagging.

Notice that with bagging we are not subsetting the training data into smaller chunks and training each tree on a different chunk. Rather, if we have a sample of size N, we are still feeding each tree a training set of size N (unless specified otherwise). But instead of the original training

data, we take a random sample of size N with replacement. For example, if our training data was [1, 2, 3, 4, 5, 6] then we might give one of our trees the following list [1, 2, 2, 3, 6, 6]. Notice that both lists are of length six and that "2" and "6" are both repeated in the randomly selected training data we give to our tree (because we sample with replacement).

Node splitting in a random forest model is based on a random subset of features for each tree.
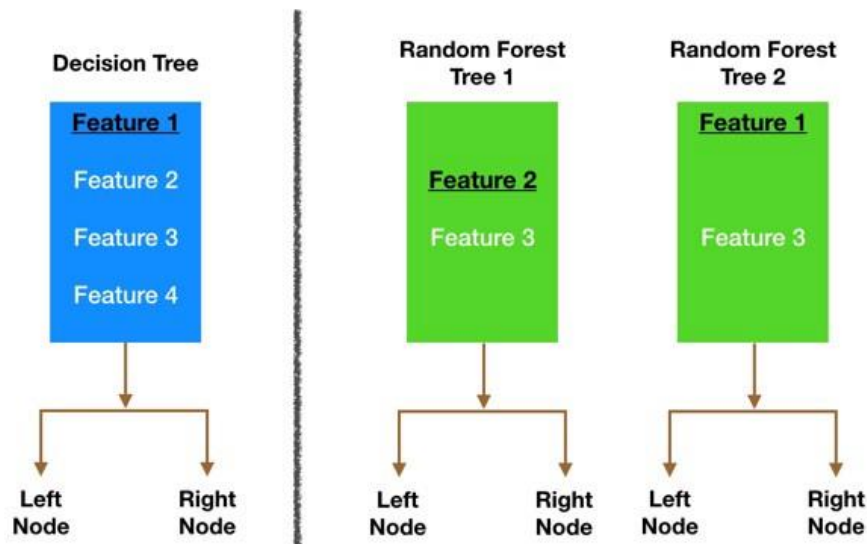
Figure 4.5: Decision Tree

Feature Randomness — In a normal decision tree, when it is time to split a node, we consider every possible feature and pick the one that produces the most separation between the observations in the left node vs. those in the right node. In contrast, each tree in a random forest can pick only from a random subset of features. This forces even more variation amongst the trees in the model and ultimately results in lower correlation across trees and more diversification.

Let's go through a visual example — in the picture above, the traditional decision tree (in blue) can select from all four features when deciding how to split the node. It decides to go with Feature 1 (black and underlined) as it splits the data into groups that are as separated as possible.

Now let's take a look at our random forest. We will just examine two of the forest's trees in this example. When we check out random forest Tree 1, we find that it it can only consider Features 2 and 3 (selected randomly) for its node splitting decision. We know from our traditional decision tree (in blue) that Feature 1 is the best feature for splitting, but Tree 1 cannot

see Feature 1 so it is forced to go with Feature 2 (black and underlined). Tree 2, on the other hand, can only see Features 1 and 3 so it is able to pick Feature 1.

So in our random forest, we end up with trees that are not only trained on different sets of data (thanks to bagging) but also use different features to make decisions.

And that, my dear reader, creates uncorrelated trees that buffer and protect each other from their errors.

Conclusion

Random forests are a personal favorite of mine. Coming from the world of finance and investments, the holy grail was always to build a bunch of uncorrelated models, each with a positive expected return, and then put them together in a portfolio to earn massive alpha (alpha = market beating returns). Much easier said than done!

5. About Dataset

5.1 Maternal Health Risk Data:

Context

Data has been collected from different hospitals, community clinics, maternal health cares through the IoT based risk monitoring system.

Age: Age in years when a woman is pregnant.

SystolicBP: Upper value of Blood Pressure in mmHg, another significant attribute during pregnancy.

DiastolicBP: Lower value of Blood Pressure in mmHg, another significant attribute during pregnancy.

BS: Blood glucose levels is in terms of a molar concentration, mmol/L.

HeartRate: A normal resting heart rate in beats per minute.

Risk Level: Predicted Risk Intensity Level during pregnancy considering the previous attribute.

Acknowledgements

Relevant Papers:

Ahmed M., Kashem M.A., Rahman M., Khatun S. (2020) Review and Analysis of Risk Factor of Maternal Health in Remote Area Using the Internet of Things (IoT). In: Kasruddin Nasir A. et al. (eds) InECCE2019. Lecture Notes in Electrical Engineering, vol 632. Springer, Singapore. [Web Link]

IoT based Risk Level Prediction Model for Maternal Health Care in the Context of Bangladesh, STI-2020, [under publication in IEEE]

Classes :

- Low risk
- Mid risk
- High risk

5.2 Gestational Diabetes Mellitus (GDM Data Set)

Gestational diabetes mellitus (GDM) is a type of diabetes (high blood sugar) that occurs in pregnant women. This study will determine whether treating pregnant women who have mild GDM improves the health of their babies. The follow-up study will examine whether factors during the previous pregnancy (such as blood sugar during pregnancy) are associated with the woman and her child's health 4-9 years later.

Classes:

- Yes
- No

5.3 Anemia Dataset

Anemia dataset containing attributes Gender, Hemoglobin, MCHC, MCV, MCH and Results. This dataset is used to predict if a patient is likely to suffer from anemia. Machine learning binary classifier algorithm to be used.

Gender: 0 - male, 1 - female

Hemoglobin: Hemoglobin is a protein in your red blood cells that carries oxygen to your body's organs and tissues and transports carbon dioxide from your organs and tissues back to your lungs

MCH: MCH is short for "mean corpuscular hemoglobin." It's the average amount in each of your red blood cells of a protein called hemoglobin, which carries oxygen around your body.

MCHC: MCHC stands for mean corpuscular hemoglobin concentration. It's a measure of the average concentration of hemoglobin inside a single red blood cell.

MCV: MCV stands for mean corpuscular volume. An MCV blood test measures the average size of your red blood cells.

Results: 0- not anemic, 1-anemic

6 SYSTEM DESIGN

System design is the first step in the development phase for any engineering product or system. The term "Design" is defined as "The process of principles for the purpose of defining a processor a system insufficient detail to permit its realization". And design is most creative and challenging phase of system development life cycle. It is an approach for the creation of the proposed system is designed, which will help in the system coding. System design is vital for efficient database management. It provides the understanding of procedural details necessary for implementing the system. A number of sub- systems are to be identified which constitute the whole system.

System design high-end decisions are taken regarding the basic system architecture, platforms and tools to be used. The system design transforms a logical representation of what a given system is required to be in the physical specification. It is an approach for the creation of the proposed system is designed, which will help in the system coding. Design starts with system requirement specification and converts it to a physical reality during the development. Important design fact or such as reliability, response time, throughput of the system maintainability, expand-ability etc. should be taken into account.

System design is the process of developing specifications for the proposed system that meet the needs established in the structured analysis. A major step in the structured design is the preparation of input and output design which will be acceptable to the user. Structured design is the process of planning a new system to replace the old one. Characteristics of well-defined system is:

(1)Acceptability

(2)Decision making ability

(3)Economy

(4)Flexibility

(5)Reliability

(6)Simplicity

6.1 Input Design

Input design converts user-oriented inputs to computer- based format, which requires careful attention. The collection of input data is the most expensive part of the system in terms of the equipment used and the number of people involved. In input design, data is accepted for computer processing and input to the system is done through mapping via some map support or links. The input design should require careful attention. In this project we use the input design as forms format. We give all the input in this forms format and the data is accepted for computer processing. All the data can be given very easily.

6.2 Output Design

Outputs are the most important a direct source of information to the user and to the department. Intelligent output design will improve the systems relationship with the user and help much in decision-making. Outputs are so used to provide a permanent hard copy of the results for later uses. Computer output is the most important and direct source of information the user. Efficient, intelligible output design should improve the systems relationship with the user and help in decision making. In this project we get the output design as forms format. All the input data given is get under processing and we get the output as forms format. These data can be easily accessible to the users.

6.3 User Interface Design

User interface design or user interface engineering is the design of user interfaces for machines and software, such as computers, home appliances, mobile devices, and other electronic devices, with the focus on maximizing usability and the user experience. The goal of user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals. Good user interface design facilitates finishing the task at hand without drawing unnecessary attention to itself. Graphic design and typography are utilized to support its usability, influencing how the user performs certain interactions and improving the aesthetic appeal of the design; design aesthetics may enhance or detract from the ability of users to use the functions of the interface. The design process must balance technical functionality and visual elements to create a system that is not only operational but also usable and adaptable to changing user needs.

7 AGILE TECHNOLOGY OVERVIEW

7.1 Introduction To Scrum

Scrum is a subset of agile. Scrum is a framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum itself is a simple framework for effective team collaboration on complex products. Scrum co-creators Ken Schwaber and Jeff Sutherland have written The Scrum Guide to explain Scrum clearly and succinctly. This Guide contains the definition of Scrum.

7.2 Principles Or Methodology Used

The SCRUM methodology is defined by team rules, events(ceremonies),artifacts and roles.

7.2.1 Scrum Team

A scrum team is a collection of individuals (typically between five and nine members) working together to deliver the requested and committed product increments. To work effectively it is important for a scrum team that everyone within the team follow a common goal. The Scrum Team share different tasks and responsibilities related to the delivery of the product. Each role are closely inter-related. It is recommended for Scrum team members work together in the same location whenever possible. There are 3 roles in a scrum team:

1. The Product Owner: The product owner is a project's key stakeholder-Usually an internal or external customer. or a scope person for the customer. There is only one product owner who conveys the overall mission and vision of the product which the team is building. The product owner is ultimately accountable for managing the product backlog and accepting completed increments of work.

2. The Scrum Master: The scrum master is the servant leader to the product owner, development team and organization. With no hierarchical authority over the team but rather more of a facilitator, The scrum master ensures that the team adheres to scrum theory, practices and rules. The scrum master protect the team by doing anything possible to help the team perform of the highest level. This may include removing impediments, facilitating meetings and helping the product owner groom the backlog.

3. The Development Team: is a self organizing, cross functional group armed with all of the skills to deliver shippable increments at the completion of each sprint.

7.2.2 Scrum Events

Scrum events are time-boxed events that means in a project, every scrum events has a predefined maximum duration. These events enable transparency on the project progress to all who are involved in the projects. The vital events of scrum are -the Sprint.

1. The Sprint: A sprint is a time-boxed period during which specific work is completed and made ready for review. Sprint are usually 2-4 weeks long but can be as short as 1 week .

2. Sprint Planning: Sprint Planning team meetings are time-boxed events that determine which product backlog items will be delivered and have the work will be achieved.

3. The daily Stand-up: The daily stand-up is a short communication meeting in which each team member quickly and transparently covers progress since the last stand-up ,planed work before the next meeting and any impediments that may be blocking his or her progress.

4. The Sprint Review: The sprint review is the "show and tell" of demonstration events for the team to present the work completed during the sprint. The product owner checks the work against predefined acceptance criteria or either accept or reject the work. The stakeholder or client give feedback to ensure that the delivered incremental must the business model.

5. Retrospective: The retrospective or retro is the final team meeting in the sprint to determine what went well, what didn't go well and how the team can improve the next sprint. Attended by the team and the scrum master, the retrospective is an important opportunity for the team to focus on its overall performance and identify startegies for continous improvement on its process.

7.2.3 Scrum Artifacts

Scrum artifacts are designed to increase transparency of information related to the delivery of the project, and provide opportunities for inspection and adaptation. They are management products useful for the creation of the specialist product of the project. There are 3 artifacts in scrum

1. Product Backlog An extended list of everything that might be needed in the final product.

2. Sprint Backlog Selected items for the product backlog to be delivered through a sprint, along with the task for delivering the item and realizing the sprint goal.

3. The Sprint Review Increment set of all the product backlog items so far in the project.

7.2.4 Scrum Rules

The rules of Agile Scrum Should be completely up to the team and governed for what works best for their processes. The best agile will tell teams to start with basic scrum events listed above and then inspect and adopt based on your teams unique needs so there is continuous improvement in the way teams work together.

7.3 Sprint

7.3.1 Product Backlog

In the simplest definition the Scrum Product Backlog is simply a list of all things that needs to be done within the project. It replaces the traditional requirements specification artifacts. These items can have a technical nature or can be user centric e.g. in the form of user stories. Product Backlog refinement is the act of adding detail, estimates, and order to items in the Product Backlog. This is an ongoing process in which the Product Owner and the development Team collaborate on the details of Product Backlog items. A Scrum product backlog contains descriptions of the functionality desired in an end product. Agile backlog prioritization is the next step. The Product Backlog is the tool used by the product owner to keep track of all of the features that stakeholders would like to see implemented in the product whereas the Sprint Backlog is a subset of the Product Backlog representing the current active Sprint iteration. However, the product owner prioritizes it.

PRODUCT BACKLOG

ID	Story	Estimation Time (Hours)	Priority
1	As a user I want to login into the system	8	1
2	As a user I want to manage my account	8	2
3	As a user I want to see my predicted maternal health risk level	8	3
4	As a user I want to check whether I have GDM or not	8	4
5	As a user I want to check whether I have Anemia or not	8	5
6	As a user I want to monitor and filter progression and health parameters	8	6

Table 1: Product backlog

7.3.2 Sprint Planner

Sprint Planning is time-boxed to a maximum of eight hours for a one-month Sprint. For shorter Sprints, the event is usually shorter. The Scrum Master ensures that the event takes place and that attendants understand its purpose. The Scrum Master teaches the Scrum Team to keep it within the time-box. The Sprint Goal is an objective set for the Sprint that can be met through the implementation of Product Backlog. It provides guidance to the Development Team on why it is building the Increment. It is created during the Sprint Planning meeting. The Sprint Goal gives the Development Team some flexibility regarding the functionality implemented within the Sprint. As the Development Team works, it does so with the Sprint Goal always in mind.

SPRINT PLANNER

Feature ID	Sprint	Start Date	End Date	Estimated Work(hrs)	Duration (Days)	Sprint Goal
01	1	16/8/22	30/8/22	8	3	Manage Login to the Application
02	2	30/8/22	11/9/22	8	3	User Management
03	3	12/9/22	25/9/22	8	3	my predicted maternal health risk level
04	4	26/9/22	10/10/22	8	3	check whether I have GDM or not
05	5	11/10/22	26/10/22	8	3	monitor and filter progression and health parameters
06	6	26/10/22	10/11/22	8	3	to check whether I have Anemia or not

Figure 2: Sprint planner

7.3.3 Ideal Burn Down Chart

A burndown chart is a graphic representation of how quickly the team is working through a customer's user stories, an agile tool that is used to capture a description of a feature from an end-user perspective. The burndown chart shows the total effort against the amount of work for each iteration. The quantity of work remaining is shown on a vertical axis, while the time that has passed since beginning the project is placed horizontally on the chart, which shows the past and the future. The burndown chart is displayed so everyone on the team can see it and is updated regularly to keep it accurate.

There are two variants that exist for a burndown chart. A sprint burndown is for work remaining in the iteration. When illustrating the work remaining for the entire project, the chart is called a product burndown. The burndown chart has several points. There's an x-axis, which is the project or iteration timeline. The y-axis is the work that needs to get done in the project. The story point estimates for the work that remains is represented by this axis.

The project starting point is the farthest point to the left of the chart and occurs on day zero of the project or iteration. The project end point is farthest to the right and marks the final day of the project or iteration.

There is an ideal work remaining line, which is a straight line connecting the start point to the end point. It shows the sum of the estimates for all the tasks that need to be completed. At the end point, the ideal line crosses the x-axis and shows there is no work left to be done. This line is based on estimates and therefore not always accurate.

Then there is the actual work remaining line that shows the actual work that remains in the project or iteration. At the beginning the actual work remaining and the ideal work remaining are the same, but as the project or iteration progresses the actual work line will fluctuate above and below the ideal work line. Each day a new point is added to this line until the project or iteration is done to make sure it's as accurate as possible. If the actual work line is above the ideal work line, it means there is more work left than originally thought. In other words, the project is behind schedule. However, if the actual work line is below the ideal work line, there is less work left than had been predicted and the project is ahead of schedule.
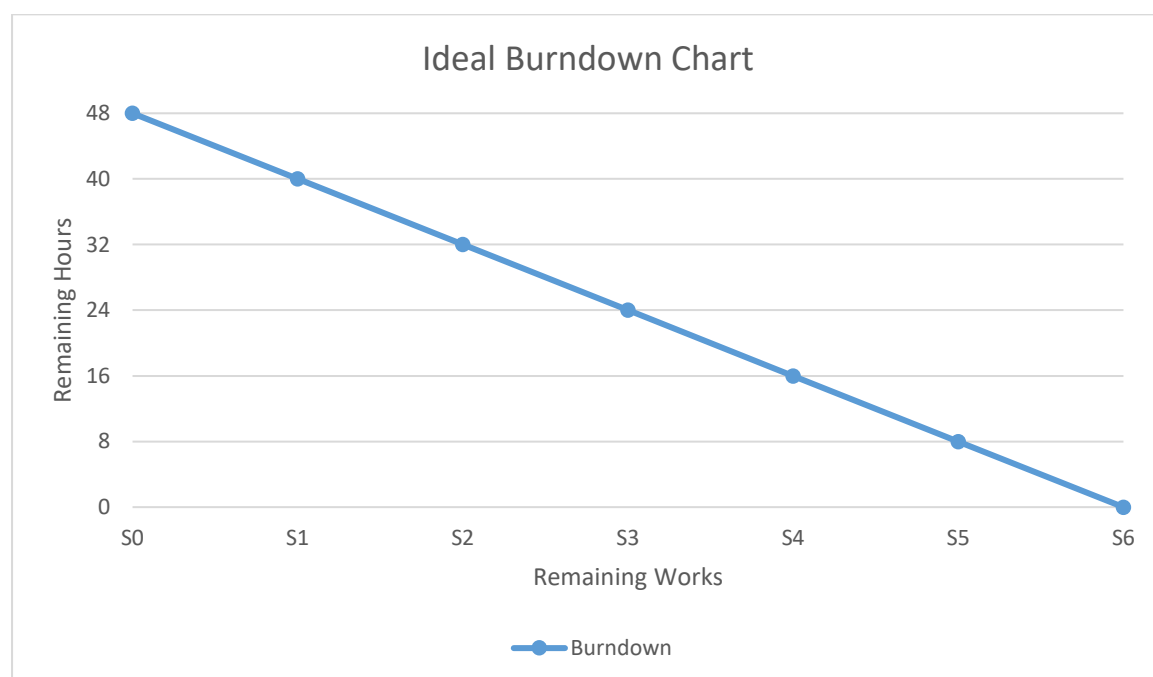

Figure 7.1: Ideal Burndown Chart

7.3.4 Git Hub Registration

GitHub is an online-browser based distributed version control system for software developers using the Git revision control system. The service provides free public repositories, issue tracking, graphs, code review, downloads, wikis, collaborator management, and more. GitHub offers free accounts for users and organizations working on public and open source projects, as well as paid accounts that offer unlimited private repositories and optional user management and security features. Git hub account creation includes the following steps:

– Go to the GitHub sign up page, then Enter a username, valid email address, and password. Use at least one lowercase letter, one numeral, and seven characters.

– Review carefully the GitHub Terms of Service and Privacy Policy before continuing and Choose a plan. Hereby anyone can finish the account creation procedure.

– You can store a variety of projects in GitHub repositories, including open source projects.

– In the upper-right corner of any page, click , and then click New repository.

– Type a short, memorable name for your repository followed by Optionally, add a description of your repository, public or private repository.

– Select Initialize this repository with a README. finally Click Create repository

– After creation, need to collaborate members by the admin.

– In the left sidebar, click Collaborators and teams.

– Under ”Collaborators”, type the name of the person you'd like to give access to the repository, then click Add collaborator.

– Next to the new collaborators name, choose the appropriate permission level: Write, Read, or Admin.

– The user will receive an email inviting them to the repository. Once they accept your invitation, they will have collaborator access to your repository.
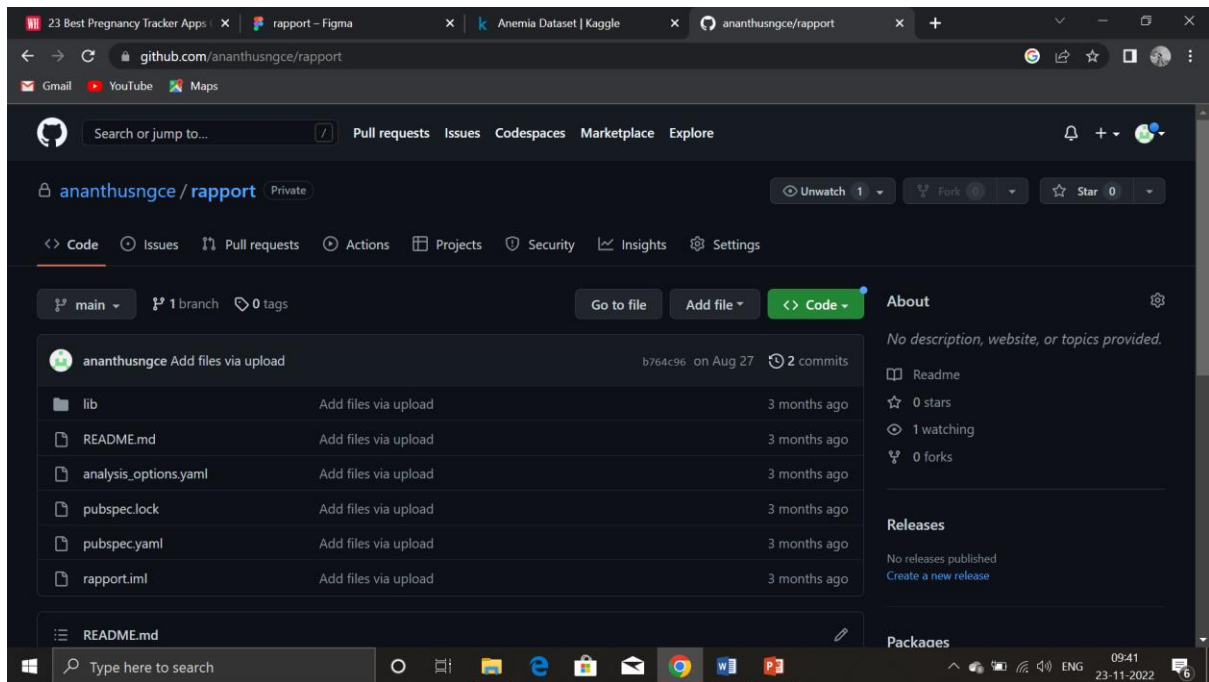
Figure 7.2: Git

8 CODING

8.1 Coding Standards Followed

The following aspects are considered to ensure good quality and high performance deliverables:

- **Naming conventions followed**

  - Classes, enums, typedefs, and extensions name should in UpperCamelCase

  - Libraries, packages, directories, and source files name should be in

  - snake_case(lowercase_with_underscores)

  - Variables, constants, parameters, and named parameters should be in lowerCamelCase.

  - Proper meaningful names should be followed.

  - Private variables names preceded with underscores

- **Proper folder structure followed**

  - Segregation of code into a proper folder structure namely providers, models, screens/pages, utils.

  - Code is properly formatted with trailing commas used appropriately.

  - Adequate comments added for documentation.

  - Remove any print statements, unused and commented code

  - Try to make code reusable with help of helper functions in utility files saved in the utils folder.

  - Widgets should also be designed to be reusable and can be saved in a widgets folder separately.

  - Avoiding static/hard coded strings in the UI screens. Constants should be derived from a single place including color codes, validation messages etc. all saved in the constants file.

- **Widget structure and usage**

- When working with infinite lists or very large lists, ListView builder is used in order to improve performance

- Split the widget into different Widgets.

- When setState() is called on a State, all descendent widgets will rebuild. Therefore, Split widget into small widgets so the setState() call rebuilds only the part of the subtree, whose UI actually needs to change.

- **Use Const in Widgets**

  - You can cache parts of your widget tree to prevent unnecessary rebuilds.

  - The easiest way is to use dart const constructors for parts of the child tree which will not change. The widget which will not change when setState is called, we should define it as constant, to prevent the widget to rebuild so it improves the Overall performance.

- **Follow Linting rules**

  - DO avoid relative imports for files in lib/. Use package imports.

  - Refer https://dart-lang.github.io/linter/lints/always_use_package_imports.html

  - And https://dart-lang.github.io/linter/lints/ for linting rules

- **Check for boundary cases and handle layout overflows properly**

  - The widgets should take care of responsiveness of the application.

  - Use of widgets which will handle screen overflows like Expanded to avoid overflow errors.

- **Build method structure**

The build method should be pure, without any side effects. This is because, it may be triggered by many external factors, such as:

- Route pop/push

- Screen resize, usually due to keyboard appearance or orientation change

- Parent widget recreated its child

- An InheritedWidget the widget depends on (Class.of(context) pattern) change

33

The build method should not have any logic to trigger an http call or modify any state.

- **Proper state management and separation of logic from UI**

    o Use provider as the recommended package for state management.

    o You can also choose to use any other approach for state management like Bloc.

    o Business logic should be separated from the UI

- **Using Provider package the right way**

    o Provide only at the needed level, instead of providing everything at the top level

    o When using Provider.of() to consume data, listen to changes only if you need to, otherwise use listen:false

    o if you want to consume the data only once. For e.g. making a method call.

    o When using Consumer widget, consume at the specific level in order to avoid rebuilding of the entire tree.

    o When using Consumer widget, use the child option to mark part of the independent widget tree which need not rebuild.

    o When using ChangeNotifierProvider, use the correct option of create or value based on if it's an existing value or creating the provider value for the first time.

- **Using Third party packages**

    o Validate any third party package being used in the application as sometimes it might break the build or not be in sync with the current flutter version. This is especially required when you are upgrading flutter, so make sure to check all your plugins and third party packages after upgrade.

Reasons for maintaining coding standards:

- People working in different modules can easily manage the source code due to familiarity.
- Acts as a blueprint for other team members to understand the code for management or changes.

- Clarity saves a lot of time doing common mistakes as it's easy to understand.
- Coding standards or industry standards improve the quality and consistency of software.

9 CONCLUSION

9.1 Future Enhancement

While talking about the future enhancements the Rapport, pregnancy monitoring and prediction app is planning to introduce few new features as future enhancements.

The features are:

- Pre -pregnancy tracking
- Post pregnancy tracking
- Exercises
- Food diets

9.2 Conclusions

Rapport apps are a booming global industry, with most pregnant women in high-income countries now using them. From the perspective of health care and health information provision, this is both encouraging and unsettling; the demand indicates a clear direction for the development of future resources, but it also underscores the importance of processes ensuring access, reliability, and quality control.

This is the first review of pregnancy app use, types of information provided, and features preferred by pregnant women in general. It indicates the demand for access to accurate information that is relevant to users, their community, and their associated health services. Given the popularity of pregnancy apps, such apps have enormous potential to be used for the provision of accurate, evidence-based health information. This app is built under enormous considering health of pregnant women. And help them to take precaution and learn about pregnancy period.

10 BIBLIOGRAPHY

- **https://flutter.dev/multi-platform/mobile**
- **https://www.tutorialspoint.com/flutter/index.html**
- **https://www.tutorialspoint.com/firebase/index.html**
- **https://docs.flutter.dev/reference/tutorials**

11 APPENDIX: GLOSSARY

11.1 Screen Shots
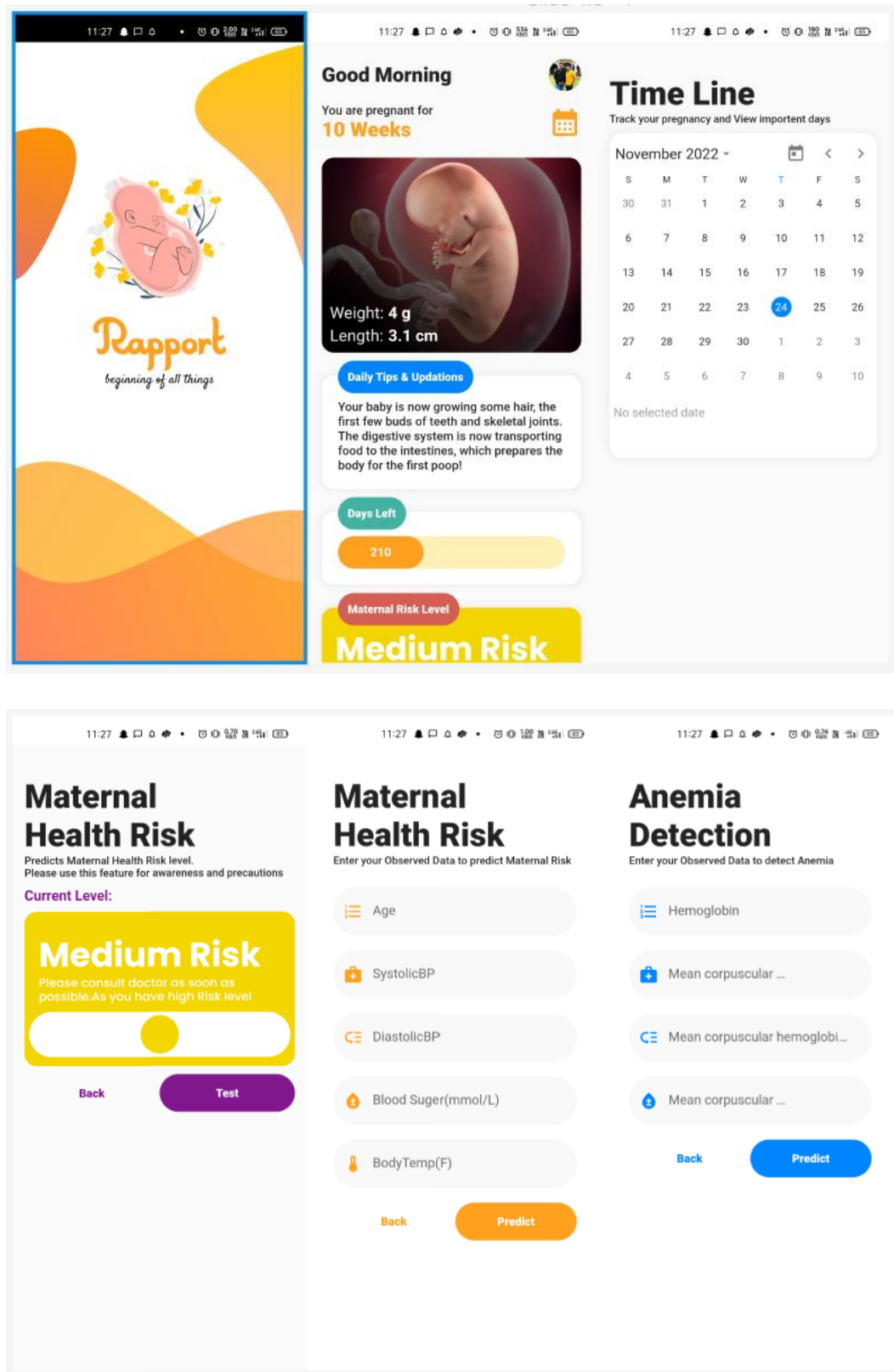
Gestational Diabetes Mellitus

Enter your Observed Data to predict Gestational Diabetes Mellitus

≡ Age

No of Pregnancy

⫤ HDL

⊞ Systolic BP

⫤ DiastolicBP

⊞ Hemoglobin

Gestation in previous Pregnancy ?

○ Yes

○ No

Hi,Susan Tom

Tap on name or photo to edit

Change Email

Change Password

Change Due Date

Change name

Sign out

Change Email

Enter new Email To Update

✉ New Email

🔒 Password

Back **Update**

Change Name

Enter new Name To Update

◎ New Name

🔒 Password

Back **Update**

Change Photo

🔒 Password

Back **Update**

Tap on Old Photo to Select

Welcome

Enter your Email and Password to login

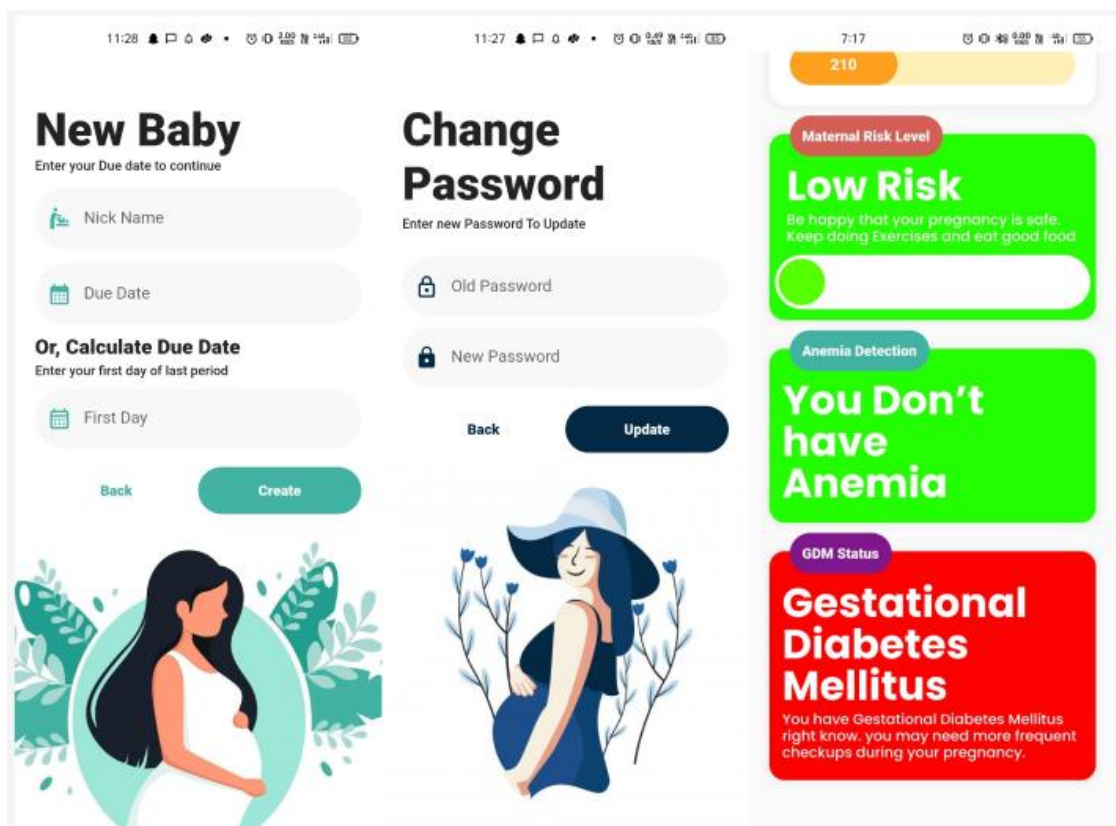✉ Your Email

🔒 Password

Sign Up **Login**

Forgot Password?
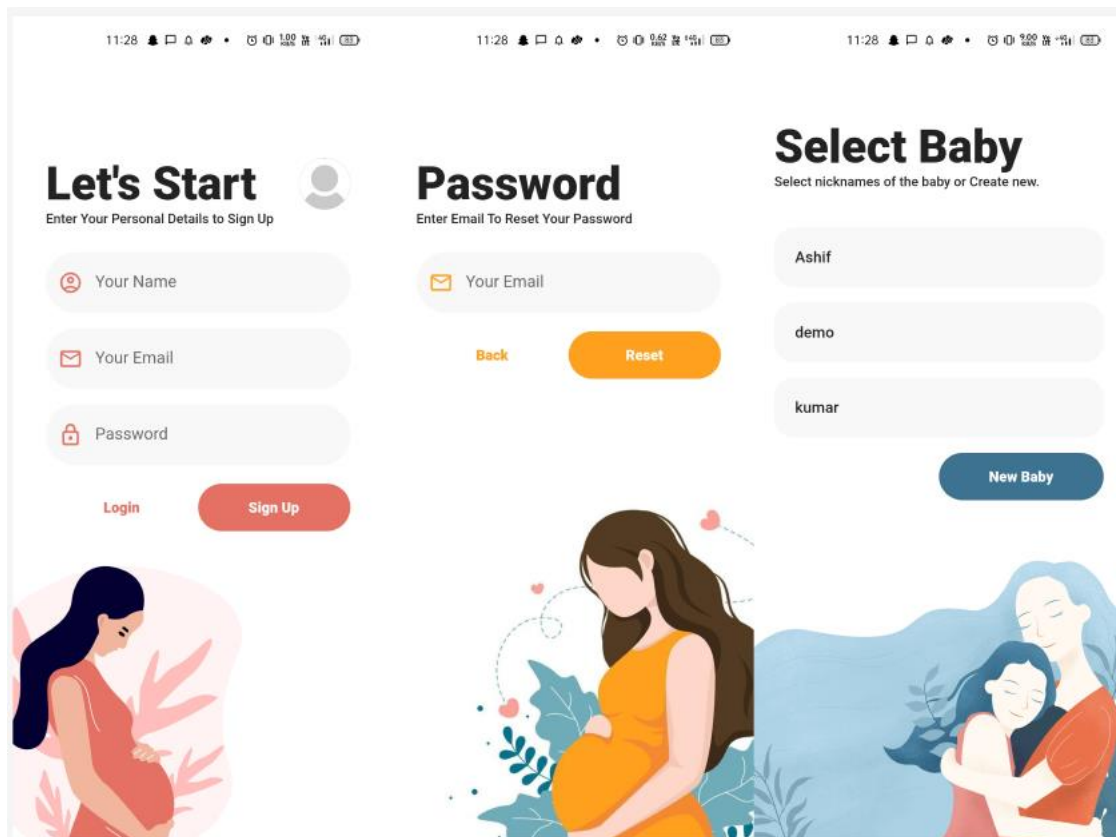
11.2 Sample Code:

Main.dart

```dart
import 'package:firebase_auth/firebase_auth.dart';

import 'package:firebase_core/firebase_core.dart';

import 'package:flutter/material.dart';

import 'package:flutter/services.dart';

import 'package:get/get_navigation/src/root/get_material_app.dart';

import 'package:rapport/screens/clist.dart';

import 'package:rapport/screens/login.dart';

import 'package:rapport/widgets/colors.dart';


void main() async {

  WidgetsFlutterBinding.ensureInitialized();

  await Firebase.initializeApp();

  runApp(const MyApp());

}


class MyApp extends StatelessWidget {

  const MyApp({Key? key}) : super(key: key);


  @override

  Widget build(BuildContext context) {

   SystemChrome.setSystemUIOverlayStyle(

     const SystemUiOverlayStyle(
```

```
      statusBarColor: Colors.white,

      statusBarIconBrightness: Brightness.dark,

    ),

  );

  return GetMaterialApp(

    debugShowCheckedModeBanner: false,

    title: 'Rapport',

    theme: ThemeData(

      primarySwatch: Palette.kToDark,

    ),

    home: StreamBuilder(

      stream: FirebaseAuth.instance.authStateChanges(),

      builder: (context, snapshot) {

        if (snapshot.connectionState == ConnectionState.active) {

          // Checking if the snapshot has any data or not

          if (snapshot.hasData) {

            // if snapshot has data which means user is logged in then we check the width of
screen and accordingly display the screen layout

            return const Childlist();

          } else if (snapshot.hasError) {

            return Center(

              child: Text('${snapshot.error}'),

            );

          }

        }
```

```dart
      // means connection to future hasnt been made yet

      if (snapshot.connectionState == ConnectionState.waiting) {

        return const Center(

          child: CircularProgressIndicator(

            color: blueColor,

          ),

        );

      }


      return LoginPage();

    },

   ),

  );

 }

}


class Palette {

 static const MaterialColor kToDark = MaterialColor(

   0xff0085FF, // 0% comes in here, this will be color picked if no shade is selected when
defining a Color property which doesn't require a swatch.

   <int, Color>{

    50: Color(0xffce5641), //10%

    100: Color(0xffb74c3a), //20%

    200: Color(0xffa04332), //30%

    300: Color(0xff89392b), //40%
```

400: Color(0xff733024), //50%

500: Color(0xff5c261d), //60%

600: Color(0xff451c16), //70%

700: Color(0xff2e130e), //80%

800: Color(0xff170907), //90%

900: Color(0xff000000), //100%

    },

  );

}

Login.dart

import 'package:flutter/material.dart';

import 'package:get/get.dart';

import 'package:http/http.dart';

import 'package:rapport/firebase/auth_methods.dart';

import 'package:rapport/screens/clist.dart';

import 'package:rapport/screens/forgetpassword.dart';

import 'package:rapport/screens/home.dart';

import 'package:rapport/screens/signup.dart';

import 'package:rapport/widgets/utils.dart';

import '../widgets/colors.dart';

class LoginPage extends StatefulWidget {

  LoginPage({Key? key}) : super(key: key);

```dart
  @override

  State<LoginPage> createState() => _LoginPageState();

}


class _LoginPageState extends State<LoginPage> {

  final TextEditingController _emailController = TextEditingController();

  final TextEditingController _passwordController = TextEditingController();

  bool _isLoading = false;


  @override

  void dispose() {

   super.dispose();

   _emailController.dispose();

   _passwordController.dispose();

  }


  void loginfunction() async {

   FocusManager.instance.primaryFocus?.unfocus();

   setState(() {

    _isLoading = true;

   });

   String res = await Authmethods().loginUser(

      email: _emailController.text, password: _passwordController.text);
```

```dart
    if (res == 'success') {

      setState(() {

        _isLoading = false;

      });

      // navigate to the home screen

      Get.offAll(

        const Childlist(),

        transition: Transition.cupertino,

      );

      showSnackBar(context, "Login Succesfully");

    } else {

      setState(() {

        _isLoading = false;

      });

      if (res ==

        "[firebase_auth/wrong-password] The password is invalid or the user does not have a
password.") {

        showSnackBar(context, "Wrong Password or Email");

      } else {

        showSnackBar(context, res);

      }

    }

  }


  @override
```

```dart
Widget build(BuildContext context) {

  return Scaffold(

    resizeToAvoidBottomInset: false,

    body: SafeArea(

      child: Container(

        alignment: Alignment.bottomCenter,

        constraints: const BoxConstraints.expand(),

        decoration: const BoxDecoration(

          image: DecorationImage(

            image: AssetImage(

              'assets/images/backgrounds/Android Large - 1.png',

            ),

            fit: BoxFit.cover,

          ),

        ),

        padding: const EdgeInsets.symmetric(horizontal: 32),

        width: double.infinity,

        child: Column(

          crossAxisAlignment: CrossAxisAlignment.center,

          mainAxisAlignment: MainAxisAlignment.start,

          children: [

            const SizedBox(height: 100),

            Container(

              alignment: Alignment.topLeft,
```

```
        child: const Text(

          'Welcome',

          style: TextStyle(

            fontSize: 45,

            fontWeight: FontWeight.w900,

          ),

        ),

      ),

      Container(

        alignment: Alignment.topLeft,

        child: const Text(

          'Enter your Email and Password to login',

          style: TextStyle(

            fontSize: 13,

            fontWeight: FontWeight.w500,

          ),

        ),

      ),

      const SizedBox(

        height: 25,

      ),

      TextField(

        controller: _emailController,

        keyboardType: TextInputType.emailAddress,
```

```dart
    decoration: InputDecoration(

     hintText: 'Your Email',

     fillColor: const Color.fromARGB(255, 248, 248, 248),

     filled: true,

     prefixIcon: const Icon(

      Icons.email_outlined,

      color: blueColor,

     ),

     border: OutlineInputBorder(

      borderSide: BorderSide.none,

      borderRadius: BorderRadius.circular(50),

     ),

    ),

   ),

   const SizedBox(height: 15),

   TextField(

    controller: _passwordController,

    autocorrect: false,

    obscureText: true,

    keyboardType: TextInputType.text,

    decoration: InputDecoration(

     hintText: 'Password',

     fillColor: const Color.fromARGB(255, 248, 248, 248),

     filled: true,
```

```dart
      prefixIcon: const Icon(

       Icons.lock_outline_rounded,

       color: blueColor,

      ),

      border: OutlineInputBorder(

       borderSide: BorderSide.none,

       borderRadius: BorderRadius.circular(50),

      ),

     ),

    ),

    const SizedBox(

     height: 18,

    ),

    Row(

     children: [

      Expanded(

       child: Material(

        color: Colors.white,

        borderRadius: const BorderRadius.all(

         Radius.circular(50),

        ),

        child: InkWell(

         borderRadius: const BorderRadius.all(

          Radius.circular(50),
```

```dart
      ),
      onTap: () {
        Get.off(
          const Signuppage(),
          transition: Transition.cupertino,
        );
      },
      child: Container(
        alignment: Alignment.center,
        padding: const EdgeInsets.all(15),
        child: const Text(
          'Sign Up',
          style: TextStyle(
            color: blueColor,
            fontWeight: FontWeight.w900,
          ),
        ),
        decoration: const ShapeDecoration(
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.all(
              Radius.circular(50),
            ),
          ),
```

```
      /*gradient: LinearGradient(

        colors: [

          Color(0xff003E87),

          Color(0xFF0075FF),


        ],

        stops: [

          0.0,

          1.0

        ],

        begin: Alignment.topLeft,

        end: Alignment.bottomRight,

        tileMode: TileMode.repeated),*/

      ),

    ),

  ),

),

Expanded(

 child: Material(

   color: blueColor,

   borderRadius: BorderRadius.circular(50),

   child: InkWell(

     splashColor: Color.fromARGB(255, 0, 89, 255),
```

```
borderRadius: BorderRadius.circular(50),

onTap: loginfunction,

child: Container(

  alignment: Alignment.center,

  padding: const EdgeInsets.all(15),

  child: !_isLoading

      ? const Text(

          'Login',

          style: TextStyle(

            color: Colors.white,

            fontWeight: FontWeight.w900,

          ),

        )

      : const SizedBox(

          child: CircularProgressIndicator(

            color: Colors.white,

            strokeWidth: 4,

          ),

          height: 16.0,

          width: 16.0,

        ),

  decoration: const ShapeDecoration(

    shape: RoundedRectangleBorder(

      borderRadius: BorderRadius.all(
```

```dart
                    Radius.circular(50),

                ),

              ),

            ),

          ),

        ),

      )

    ],

  ),

  const SizedBox(

    height: 10,

  ),

  InkWell(

    onTap: () {

      Get.to(

        const ForgetPassword(),

        transition: Transition.cupertino,

      );

    },

    splashColor: blueColor,

    child: Container(

      color: Colors.transparent,

      alignment: Alignment.centerRight,
```

```dart
            child: const Text(

             'Forgot Password?',

             style: TextStyle(

              color: blueColor,

              fontSize: 15,

              fontWeight: FontWeight.w500,

             ),

            ),

           ),

          ),

         ],

        ),

       ),

      ),

     );

    }

}
```

11.3 DATABASE DESGIN

users
email
PhotoUrl
uid
username

Prediction
Anemia
MaternalRisk
GDM

child
childern

dailyupdate
meg

babydetails
week
lenght
Height

11.4 Actual Burn down Chart

Actual Burndown Chart