Technical Report

Real-Time AI-Powered Health Coaching for Diabetes: Prototype

Team Name: Data Divers (Team 1)

Team Members: Ashna Ali, Sreevardhan Reddy Soma, Venkata Sai Veeramalla, Sai Kiran Basetty

9/23/2024

**Project Title:** Real-Time AI-Powered Health Coaching for Diabetes

**Problem Statement**

Diabetes is a widespread chronic condition, with its forms—Type 2 diabetes, prediabetes, and gestational diabetes—posing significant health risks if not detected and managed early. Accurate prediction of diabetes risk remains a challenge due to the complexity of contributing factors, including demographic, biometric, and lifestyle variables, as well as real-time physiological data such as glucose levels. Existing prediction models often rely on structured health data but fail to incorporate continuous glucose monitoring (CGM) data, which can provide critical insights into real-time glucose fluctuations. Additionally, imbalanced datasets, particularly with underrepresented classes like gestational diabetes and prediabetes, compromise the reliability of predictive models.

This study aims to address these challenges by developing an advanced machine learning model that integrates both structured data (e.g., age, BMI, blood pressure) and time-series CGM data to improve the prediction accuracy of diabetes risk. The project leverages techniques such as Synthetic Minority Over-sampling (SMOTE) to handle data imbalance and Recursive Feature Elimination (RFE) to select the most relevant predictors. To ensure interpretability, Local Interpretable Model-agnostic Explanations (LIME) are employed to provide transparency into the model’s predictions. The model is deployed through a user-friendly Streamlit interface, allowing real-time interaction and prediction, with results stored in a secure MongoDB repository.

This research seeks to enhance early detection of diabetes and its subtypes by addressing key limitations in current predictive methodologies, thus contributing to more effective disease management and intervention.

**Objectives**

1. **Develop a machine learning model** capable of predicting the likelihood of pre-diabetes, Type 2 diabetes, or gestational diabetes using a combination of user demographic data, biometric measurements, and continuous glucose monitoring (CGM) time-series data.

2. **Gather and preprocess a comprehensive dataset** that includes demographic information, lifestyle factors, biometric measurements, and CGM data relevant to diabetes prediction (i.e., age, weight, glucose levels, blood pressure, gender, sleep patterns, etc.). Data preprocessing includes handling missing values, outliers, and applying normalization and encoding techniques.

3. **Train, fine-tune, and validate the model** using advanced machine learning techniques, including transfer learning and Recursive Feature Elimination (RFE), to improve accuracy, sensitivity, and specificity while addressing data imbalance through techniques such as SMOTE.

4. **Integrate the model into a user-friendly Streamlit-based web application**, allowing real-time user interaction and prediction submissions. The interface should be easily accessible by healthcare providers or individuals.

5. **Implement a secure MongoDB database** for storing user input, CGM data, and model predictions, enabling real-time updates and retrieval.

6. **Evaluate the model’s performance** against existing diagnostic methods, ensuring the model’s effectiveness, interpretability (via LIME), and reliability in predicting diabetes-related outcomes.

**Technical Details**

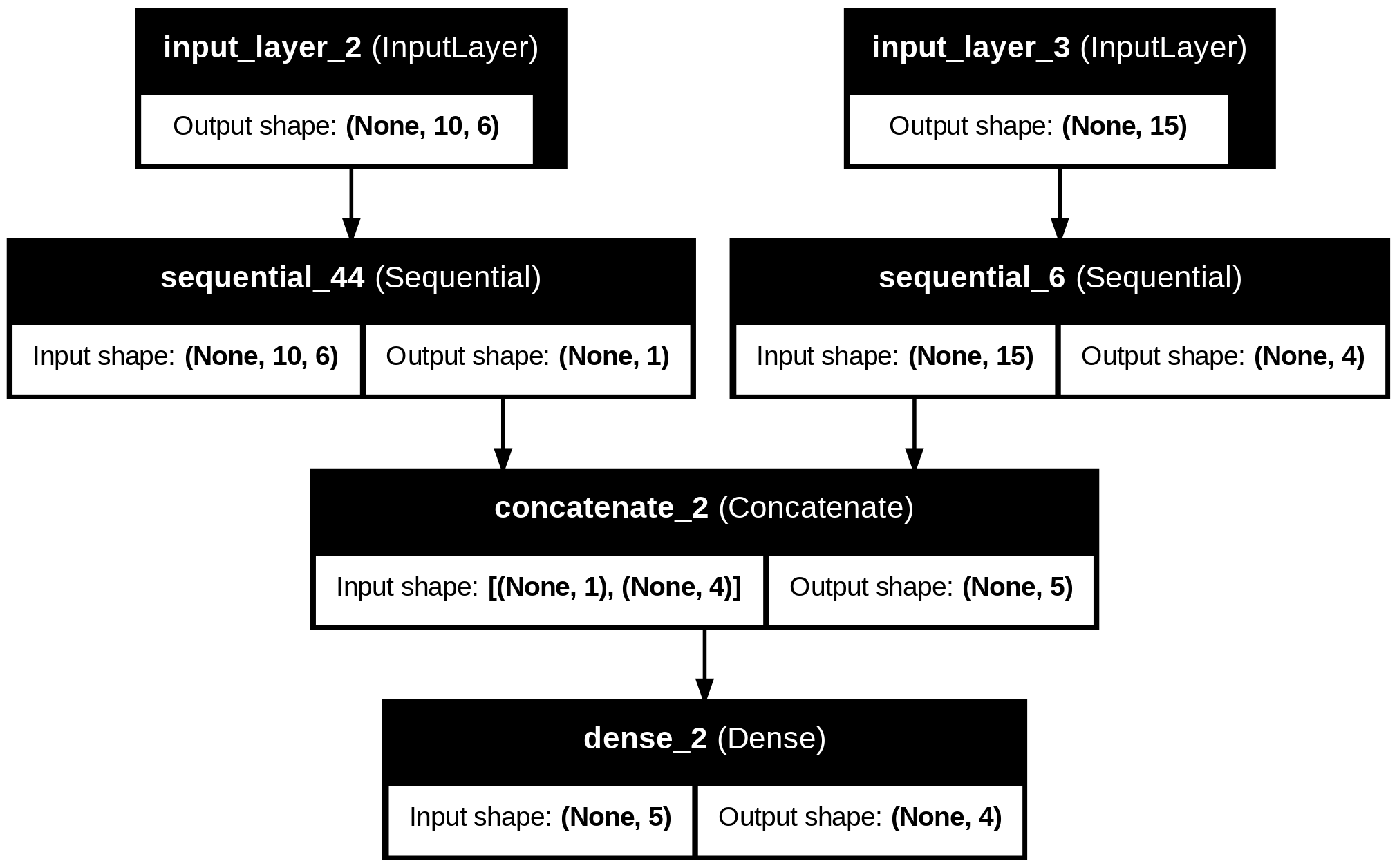
We employed a machine learning approach using a comprehensive dataset that incorporated user demographics and biometric data. The main features include age, BMI, sleep patterns, gender, and blood pressure levels. Preprocessing involved cleaning the data by filling missing values, handling outliers, and applying normalization and encoding techniques. The numerical features (age, BMI, blood pressure, etc.) were scaled using the standard scaler and the categorical features (gender, family history of diabetes, high blood pressure diagnosis, etc.) were one hot encoded.

The deep learning model was built using **TensorFlow and Keras**, focusing on improving prediction accuracy through a well-curated set of features. Initially, a deep neural network (DNN) was developed, and feature importance was identified to reduce model complexity. This reduction allowed us to focus on the top predictors of diabetes risk without compromising model performance. In our original dataset, smoking history, alcohol usage, and insulin levels were also features but were eliminated when applying feature importance.

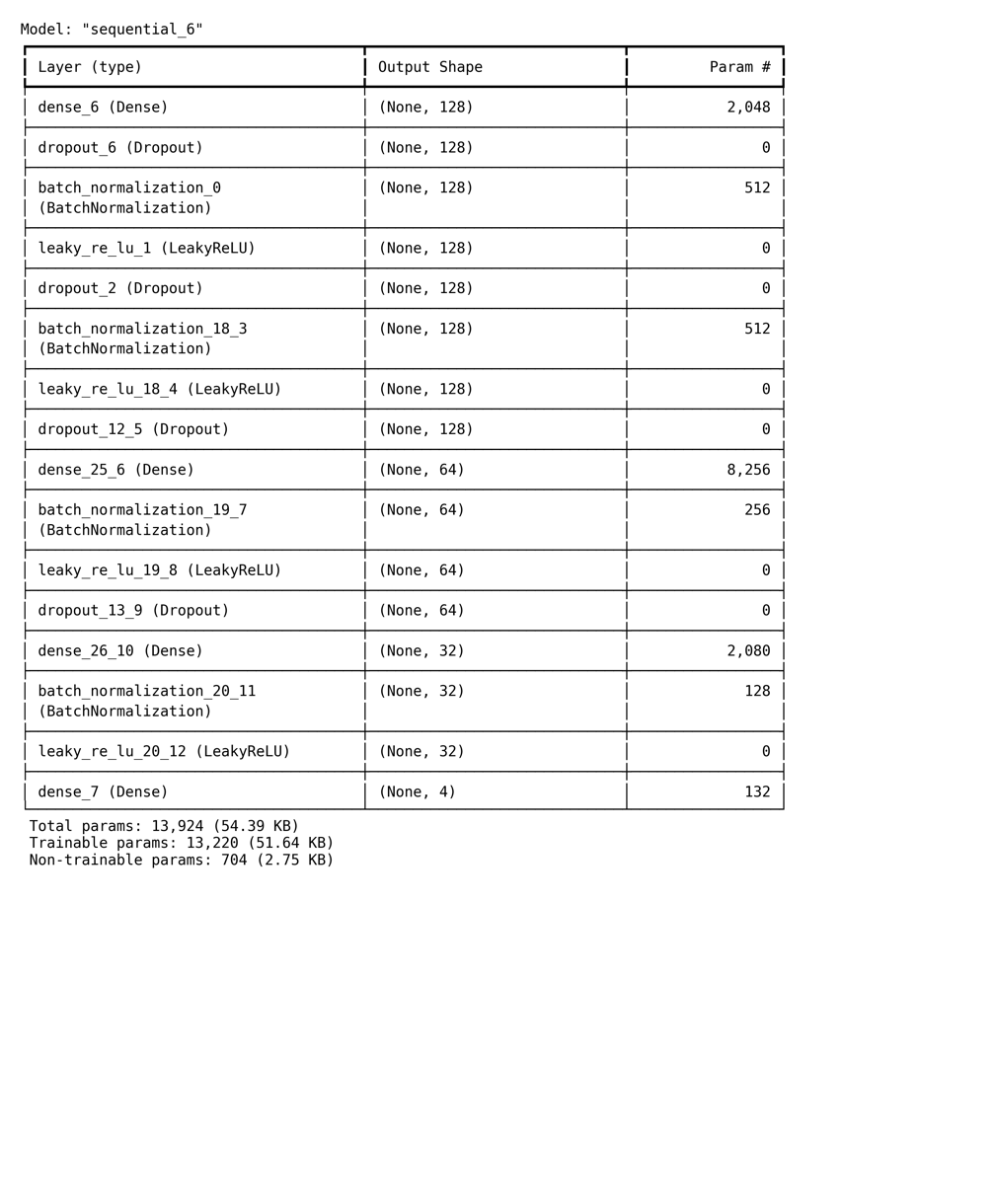
The model itself was trained on several datasets with some being sourced from Kaggle and GitHub repositories. We have uploaded those datasets as a binary file release on our team GitHub repository. Our model was initially trained on a dataset that had mainly health indicators such as high blood pressure, high cholesterol, physical activity, etc. The target variable was only for diagnosis of type 2 diabetes, prediabetes, or no diabetes. So, to include more samples of women diagnosed with gestational diabetes, we fine-tuned, and transfer learned on the PIMA Indians Dataset (which is labeled as gestational\_diabetes\_dataset on Github). We used the same technique to gather more samples of prediabetes and type 2 diabetes as well with other datasets we found from other sources.

Finally for our time series data, we fine-tuned, and transfer learned on a dataset that we found on the GitHub repository Awesome-CGM. These datasets contain continuous glucose monitoring data. Since this dataset is primarily used to classify patients with and without type 2 diabetes, the other 2 classes of prediabetes and gestational diabetes were omitted. To train the model, we had to divide our model into two sections; one section deals with the structured data of the health indicators and symptoms mentioned before, and the other part of the model handles the time series data. The model topology is shown below:

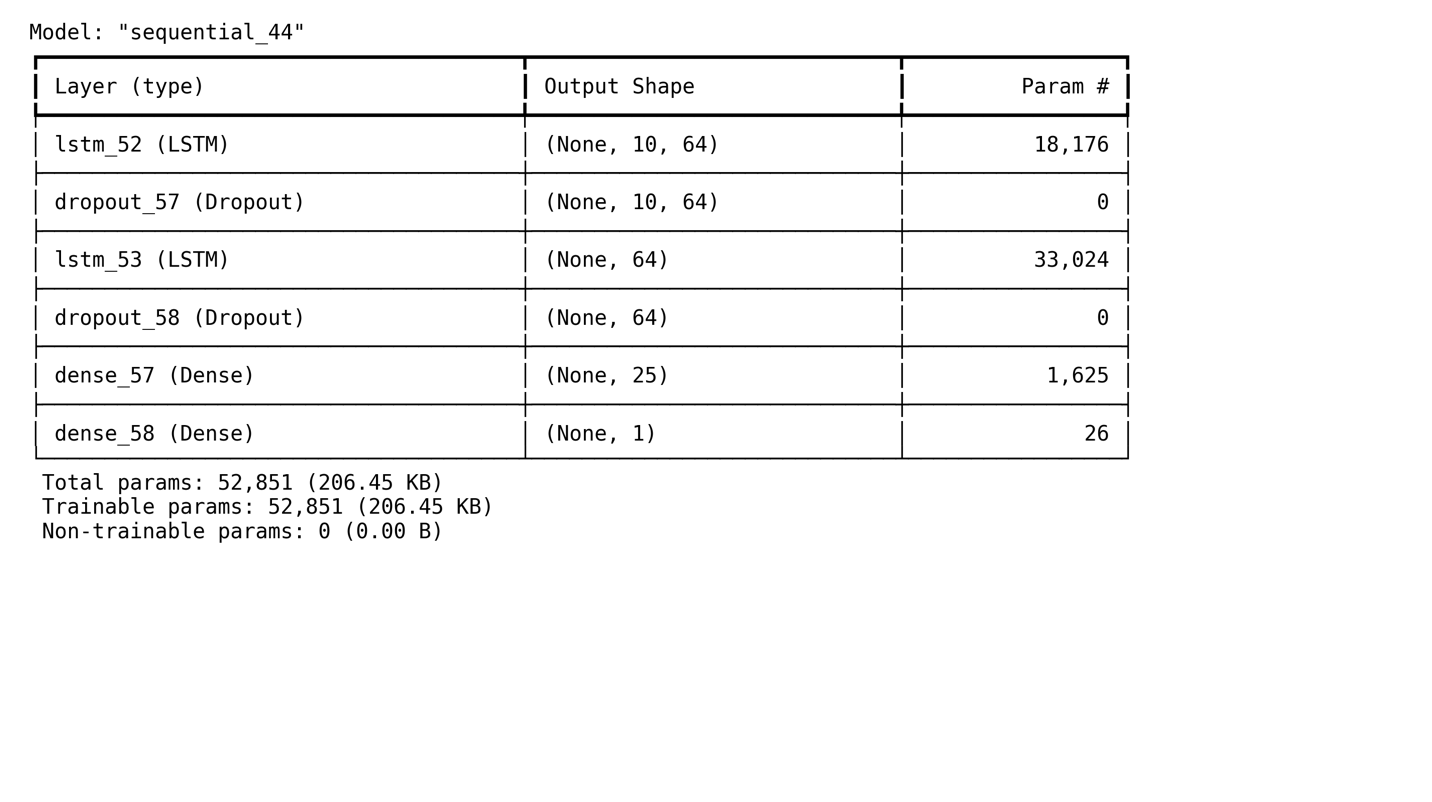
**Final Model Topology**



***Structured Data Model (Sequential 6)***



***Time Series Model (Sequential 44)***



The model was validated using cross-validation and hyperparameter tuning techniques, achieving high accuracy in predicting diabetes risk categories (pre-diabetes, Type 2 diabetes, or gestational diabetes). Model performance was evaluated using metrics such as ROC-AUC, F1-score, and confusion matrices.

**Triangle Model:** Frontend, Data Repository, and AI/ML Component

The implementation of the project can be broken down into three key components:

1. **Frontend**: a web-based user interface was developed using the Streamlit framework, providing an accessible interface for users to input their health data. Users can view the prediction after submitting the health data in a survey question-based frontend. The questionnaire consists of these prompts:
   1. How many pregnancies have you had in total?
   2. Enter your height in inches.
   3. Enter your weight in inches.
   4. How many hours of sleep were you staying still in bed (not interrupted sleep)?
   5. How many hours of sleep per day do you get on average?
   6. Select your gender.
   7. Enter your age.
   8. Do you have a family history of diabetes?
   9. On average, how physically active are you per day?
   10. What is your blood pressure level?
   11. Are you diagnosed with high blood pressure?
   12. Enter your CGM (Continuous Glucose Monitoring) data.

After the user clicks submit, the predictions are calculated and sent to the MongoDB database directly. Streamlit offers a “no API necessary” structure to send data to the database.

1. **Data Repository**: The dataset, hosted on **MongoDB**, includes user demographics, lifestyle information, and clinical metrics. This repository is securely integrated with the AI component, allowing real-time data access and updates. The predictions are sent to the dataset as well along with the user input. The database entry is represented as a JSON document, which includes various attributes related to input values, continuous glucose monitoring (CGM) data, prediction probabilities, and a diagnosis. The key components of this structure are:
   1. **\_id**:
      1. A unique identifier for the database entry. It uses MongoDB’s ObjectId format, where each entry is assigned a unique identifier ($oid), e.g., "66edc8d89d71445bfcb518a4".
   2. **input\_values**:
      1. A dictionary that contains the input values used for prediction. Each input is stored in either an integer ($numberInt) or floating-point ($numberDouble) format. The input values include:
         1. pregnancies: Number of pregnancies, stored as an integer.
         2. bmi: Body Mass Index (BMI), stored as a double.
         3. sound\_sleep: Hours of sound sleep, stored as a double.
         4. sleep: Total sleep duration in hours, stored as a double.
         5. gender\_male and gender\_female: Binary indicators (1 or 0) representing the gender of the individual.
         6. age: The age of the individual, stored as an integer.
         7. family\_diabetes\_yes and family\_diabetes\_no: Binary indicators of family history of diabetes.
         8. physically\_active\_less\_than\_half\_hr and physically\_active\_none: Binary indicators of physical activity levels.
         9. bp\_level\_high and bp\_level\_normal: Indicators of blood pressure level.
         10. high\_bp\_no and high\_bp\_yes: Indicators of whether the individual has high blood pressure.
   3. **cgm\_data**:
      1. Continuous Glucose Monitoring (CGM) data represented as a time-series. Each entry contains timestamped glucose values (e.g., "0, 87.0, 25.4, 77, 6.3, 413"), where the first number indicates the timestamp and the following values indicate glucose-related metrics.
   4. **prediction**:
      1. A nested array containing the probabilities of various conditions. Each element is represented as a floating-point number ($numberDouble). These probabilities correspond to the likelihood of different diabetes-related outcomes, such as:
         1. The risk of prediabetes.
         2. The risk of type 2 diabetes.
         3. Other conditions.
   5. **diagnosis**:
      1. The final diagnosis derived from the model predictions. In this case, the diagnosis is "No diabetes".
2. **AI/ML Component**: The machine learning model, imported from the local system, performs real-time predictions based on user input. Model predictions are sent back to the frontend for user interaction.

**Challenges and Solutions**

1. **Data Imbalance**: One of the major challenges was handling class imbalance in the dataset, where most of the datasets represented non-diabetic individuals. This was addressed by applying techniques such as **Synthetic Minority Over-sampling Technique (SMOTE)** to generate synthetic data for underrepresented classes, improving model performance.
2. **Feature Selection**: Selecting the most relevant features for the model was crucial. By applying **Recursive Feature Elimination (RFE)**, we narrowed down the features to the top 15 most significant factors affecting diabetes risk, such as age, BMI, and family history.
3. **Model Interpretability**: The deep learning model’s black-box nature posed challenges in explaining its predictions. To address this, we used **LIME (Local Interpretable Model-agnostic Explanations)** to generate interpretable results, making the model’s predictions more transparent and understandable to end users.

**Next steps**

**1.Model building methodology**

***a) Implementation of* Dual *model***

In the current project, the application interface allows both male and female users to predict their diabetes risk by selecting their gender from a dropdown menu. This selection customizes the user experience by displaying input fields relevant to each gender, ensuring that data collection is personalized. Presently, a single unified model is used for predicting diabetes risk for both males and females, processing inputs uniformly despite gender-specific differences. However, our future approach aims to enhance prediction accuracy and personalization by implementing two distinct models: one specifically fine-tuned for male-related characteristics and another for female-specific factors. For example, the male model would focus on general attributes like weight, height and BMI along with the other features that used to predict diabetes, while the female model would consider factors like pregnancy history along with similar factors of Male that significantly affect diabetes risk.

***b) Benefits***

The benefits of using two models are multifold. Gender-specific models enable more accurate and relevant predictions by focusing on the most impactful features for each gender, reducing noise and enhancing model performance. This approach also improves the interpretability of results, providing clearer insights into diabetes risk factors unique to males and females, thus supporting more targeted feedback and interventions. By addressing gender-specific features metabolic rates in males, and pregnancy in females—these models offer a tailored risk assessment that enhances the user’s understanding of their health.

Furthermore, implementing gender-specific models establishes a scalable framework that can be expanded in the future to include additional models for specific subgroups, continually adapting and improving the prediction process. This innovative approach not only personalizes diabetes risk prediction but also aligns predictive algorithms with the unique health profiles of male and female users, ultimately contributing to better disease management and health outcomes.

***c)* *Implementation of* Retraining *the model with user data*** (Either automation or traditional approach)

Re-training the model with user data is a powerful approach to enhancing its efficiency and accuracy, especially when the initial training data is limited or lacks diversity. In the current project, user input data—such as demographic, biometric, and CGM readings—are stored securely in a data repository like MongoDB. Each time a user interacts with the application, a new row of data is created and saved, capturing the relevant features for diabetes prediction. This user-generated data is then periodically merged with the existing training dataset, ensuring the combined dataset reflects real-world conditions and user-specific characteristics. Preprocessing steps, including normalization, encoding, and outlier handling, are applied to maintain the quality of the data before it is used to re-train the model. Re-training allows the model to learn from new patterns, adapting to shifts in user demographics and emerging health trends, which ultimately improves its predictive performance.

Currently, the re-training process may be triggered manually, but future improvements aim to automate this workflow, creating a continuous learning loop. Automation involves setting up data pipelines that regularly integrate new user data and initiate re-training cycles, ensuring that the model remains up to date with the latest information. This can be achieved through tools like Apache Airflow or cloud-based solutions that streamline data integration and model updates. The primary benefits of this approach include enhanced model accuracy and adaptability, as the model continually learns from the evolving user data, capturing nuances that static datasets often miss. By integrating user-specific data into the training process, the model delivers more personalized and relevant predictions, ultimately supporting better diabetes management and outcomes.

**2. Automating the process of saving the data & convert it to useful data format**

To enhance the re-training process, we are developing an automated system to convert user input data, initially stored as JSON in the data repository, into a more useful format like CSV or Excel. When users enter their data into the application, it is saved in JSON format, which is not directly compatible with traditional datasets used for model training. By converting this data into CSV or Excel, it can be easily merged with existing datasets, facilitating smoother integration for re-training the model. Automating this conversion process to run periodically will streamline the workflow, ensuring that new user data is consistently prepared and ready for model updates without manual intervention. This approach not only saves time but also ensures that the model benefits from continuous, up-to-date data, enhancing its overall accuracy and responsiveness to new patterns observed in user inputs.

**3. Developing the Recommendations based system to prediction:**

The next step in the project involves developing a recommendation-based system to provide personalized guidance and coaching for diabetes management. When a user is identified as having pre-diabetes, Type 2 diabetes, or gestational diabetes, the system will offer tailored recommendations on food choices, lifestyle habits, and other preventive measures to help manage and control their condition. For users who are non-diabetic, the system will provide preventive suggestions to help reduce the risk of developing diabetes in the future. This feature aims to enhance the user experience by offering actionable advice based on individual health profiles, supporting better long-term health outcomes. Currently, the model is deployed, and connections are functioning smoothly, laying the groundwork for integrating this recommendation feature. Developing and incorporating this component will add significant value to the project, transforming it into a comprehensive tool for diabetes prevention and management.

**4. Idea to automate the process of gathering time-series data (CGM data) as input from external application:**

To streamline the process of collecting time-series data, particularly Continuous Glucose Monitoring (CGM) data, we are exploring ways to automate data input directly from external applications. Currently, users are required to manually enter up to 60 CGM values, which can be cumbersome, prone to errors, or even lead to missing data, especially for older users. To address this, we are developing a feature that will automatically pull CGM data from compatible applications on the user’s phone, eliminating the need for manual entry. This automation will significantly improve data accuracy, reduce user burden, and ensure a seamless integration of real-time glucose readings into our model. Although this feature is still in the initial stages and under discussion with the team, its implementation will enhance the overall efficiency and reliability of our diabetes prediction and management system.

**5. Creating an interactive UI:**

Creating an interactive user interface (UI) is crucial for providing a convenient and engaging user experience, especially for applications focused on health management like ours. A well-designed UI enhances usability, making it easier for users to input data and navigate the application. While we are currently developing a basic UI to meet project deadlines, our future plans include implementing a more advanced and interactive interface tailored specifically for male and female users. This gender-specific UI will dynamically adjust to display relevant fields based on the user’s selection, ensuring that the data collection process is personalized, intuitive, and efficient. By focusing on enhancing the UI, we aim to improve user satisfaction and encourage more accurate data input, ultimately supporting better diabetes management and prediction outcomes.