**1. Project Overview**

MyHealthTracker is a personal health management application that allows users to input test results and personal data, track trends over time, compare with similar demographic groups, and predict disease risks using machine learning models. The application aims to provide users with personalized health insights and recommendations based on their lifestyle and medical data.

**2. Datasets Used and Data Adaptation**

The datasets were carefully chosen and adapted to align with the database schema used in MyHealthTracker. The main considerations in selecting and transforming the datasets include ensuring consistency in data format, normalizing values to match predefined categories, and handling missing or inconsistent values effectively.

The following external datasets were utilized to build predictive models:

* **Health Checkup Result Dataset** ([Kaggle](https://www.kaggle.com/datasets/hongseoi/health-checkup-result)) - Used for general health metrics.
* **Heart Disease Dataset** ([Kaggle](https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset)) - Adapted to match heart disease risk factors present in MyHealthTracker.
* **Diabetes Health Indicators Dataset** ([Kaggle](https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset)) - Adjusted to map blood pressure, cholesterol, BMI, and lifestyle indicators.
* **Stroke Prediction Dataset** ([Kaggle](https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset)) - Modified to fit MyHealthTracker’s classification needs for stroke prediction.
* **Depression Dataset** ([Kaggle](https://www.kaggle.com/datasets/anthonytherrien/depression-dataset)) - Processed to match user lifestyle factors stored in the database.

**Data Transformation & Feature Engineering**

Each dataset underwent transformations to align with the database fields and maintain consistency across all models:

* **Handling Missing Values:** Missing numerical values were replaced with mean values or interpolated from similar samples. Categorical missing values were mapped to “Unknown” or the most common category.
* **Feature Mapping:** Fields were remapped to match MyHealthTracker’s database format. For example:
  + Age groups were converted into categorical values ranging from 1-13.
  + Lifestyle attributes such as smoking, drinking, and physical activity were standardized to match the options in MyHealthTracker’s schema.
* **Scaling & Normalization:**
  + MinMaxScaler was used for datasets where absolute values mattered (e.g., BMI, glucose levels).
  + StandardScaler was applied to datasets where relative differences were more significant (e.g., cholesterol, blood pressure readings).
* **Class Imbalance Handling:**
  + The Stroke and Depression datasets had a significant imbalance in positive class labels, requiring SMOTE (Synthetic Minority Over-sampling Technique) to improve model performance.
* **Feature Engineering:**
  + BMI was calculated from height and weight when necessary.
  + A categorical variable for education levels was mapped from dataset-specific values to align with MyHealthTracker’s scale.
  + Derived variables, such as a computed Hypertension score, were added to improve prediction accuracy.
  + Blood sugar levels were transformed into categories based on medical classification.

**3. Model-Specific Data Adaptation and Evaluation**

Each machine learning model required specific data transformations to align with MyHealthTracker’s schema. Below is a detailed breakdown of models evaluated, their performance, the chosen best-performing model, and feature descriptions.

**3.1 Heart Disease Model**

* **Features:**
  + Age: Patient's age in years.
  + Gender: Coded as 1 (Male) or 2 (Female).
  + Blood Pressure (BP): Categorized into normal (≤120/80 mmHg) or high (>120/80 mmHg).
  + Cholesterol Levels: Mapped into risk categories, high if total cholesterol > 200 mg/dL.
  + Fasting Blood Sugar (FBS): Binary indicator, 1 if >126 mg/dL, 0 otherwise.
* **Models Evaluated:**
  + Logistic Regression
  + Random Forest Classifier
  + Gradient Boosting Classifier
  + Neural Network (Deep Learning)
* **Best Performing Model:** Gradient Boosting
* **Impact:** Improved classification accuracy due to structured risk assessment.

**3.2 Stroke Prediction Model**

* **Features:**
  + Age: Patient’s age in years.
  + Gender: Mapped to 1 (Male), 2 (Female), or 3 (Other).
  + Hypertension: Binary indicator, 1 if systolic BP >140 mmHg or diastolic BP >90 mmHg.
  + Smoking Status: Coded as never smoked (0), formerly smoked (1), or currently smoking (2).
  + Average Glucose Level: Normalized continuous variable.
  + BMI: Patient’s Body Mass Index.
* **Models Evaluated:**
  + Logistic Regression
  + Random Forest Classifier
  + Gradient Boosting Classifier
  + XGBoost
  + Neural Network
* **Best Performing Model:** XGBoost
* **Impact:** Improved feature consistency leading to better prediction results.

**3.3 Diabetes Prediction Model**

* **Features:**
  + High Blood Pressure: Binary indicator, 1 if BP >140/90 mmHg, 0 otherwise.
  + High Cholesterol: Binary indicator, 1 if total cholesterol > 200 mg/dL, 0 otherwise. 1 if cholesterol > 200 mg/dL, 0 otherwise.
  + BMI: Continuous measure of body mass index.
  + Smoking Status: Binary indicator, 1 if currently smoking, 0 otherwise.
  + Physical Activity: Categorized as low (0), moderate (1), or high (2).
  + Alcohol Consumption: Binary indicator, 1 if heavy drinker, 0 otherwise.
  + Gender: Mapped based on MyHealthTracker’s schema.
  + Age Group: Mapped to predefined database categories.
  + Education Level: Coded from high school (1) to PhD (5).
* **Models Evaluated:**
  + Logistic Regression
  + Random Forest Classifier
  + Gradient Boosting Classifier
  + XGBoost
  + Neural Network
* **Best Performing Model:** XGBoost
* **Impact:** Enhanced compatibility with MyHealthTracker’s schema, making predictions more actionable.

**3.4 Depression Prediction Model**

* **Features:**
  + Age: Patient’s age in years.
  + Marital Status: Mapped into predefined categories.
  + Education Level: Mapped to database categories.
  + Number of Children: Numeric count of dependents.
  + Smoking Status: Coded as non-smoker (0), former smoker (1), current smoker (2).
  + Physical Activity: Coded as low (0), moderate (1), or high (2).
  + Work Status: Binary indicator, 1 if employed, 0 otherwise.
  + Alcohol Consumption: Binary indicator, 1 if heavy drinker, 0 otherwise.
  + Dietary Habits: Categorized from healthy (0) to unhealthy (2).
  + Sleep Patterns: Mapped into categories based on severity.
* **Best Performing Model:** Neural Network
* **Impact:** Better alignment with MyHealthTracker’s user data, improving the model’s interpretability.

**4. Application API Endpoints**

The MyHealthTracker application provides API endpoints through a Flask-based backend, enabling real-time health risk predictions based on user data. Each endpoint interacts with the MySQL database, retrieves necessary user information, processes relevant features, loads the most effective trained model, and returns a risk assessment along with a probability score.

**4.1 Heart Disease Prediction API**

* **Endpoint:** /api/predict\_heart\_disease
* **Functionality:** Retrieves user health data, including age, gender, blood pressure, cholesterol levels, and fasting blood sugar. It processes these features, applies the trained Gradient Boosting model, and provides a heart disease risk assessment.
* **Response:**
  + risk\_level: Categorized as low, moderate, or high risk.
  + probability: The likelihood (in percentage) of heart disease presence.

**4.2 Stroke Prediction API**

* **Endpoint:** /api/predict\_stroke
* **Functionality:** Uses demographic and health factors such as age, gender, hypertension status, smoking habits, BMI, and glucose levels to predict stroke risk. The trained XGBoost model generates an accurate risk assessment.
* **Response:**
  + risk\_level: Low or high stroke risk.
  + probability: Percentage chance of stroke occurrence.

**4.3 Diabetes Prediction API**

* **Endpoint:** /api/predict\_diabetes
* **Functionality:** Retrieves lifestyle and medical test data, including blood pressure, cholesterol, BMI, physical activity, alcohol consumption, and smoking habits. It applies the trained XGBoost model to classify the user’s diabetes risk.
* **Response:**
  + risk\_level: Low, moderate, or high diabetes risk.
  + probability: Percentage likelihood of diabetes diagnosis.

**4.4 Depression Prediction API**

* **Endpoint:** /api/predict\_depression
* **Functionality:** Uses user-reported lifestyle and demographic data, such as marital status, education, smoking status, alcohol consumption, dietary habits, and sleep patterns. The trained Neural Network model predicts the user’s likelihood of suffering from depression.
* **Response:**
  + risk\_level: Low or high depression risk.
  + probability: Probability score indicating the likelihood of depression.

These API endpoints ensure seamless integration with the MyHealthTracker front-end, providing users with real-time, personalized health insights based on their input data.

**.5Models Prediction Directory Structure**

The **models prediction** directory contains various files related to machine learning models used for predicting health risks. Below is a structured breakdown of the contents:

**.51. Model Files (.pkl)**

These files store trained machine learning models, allowing them to be loaded and used for predictions without retraining:

* **best\_diabetes\_model\_Gradient Boosting.pkl** – The best-performing diabetes prediction model using Gradient Boosting.
* **best\_heart\_disease\_model.pkl** – The best heart disease prediction model .
* **best\_logistic\_regression\_model\_depression.pkl** – A logistic regression model trained for depression prediction.
* **best\_stroke\_model\_Gradient Boosting.pkl** – The best-performing stroke prediction model using Gradient Boosting.

**.52. Python Scripts (.py)**

These scripts contain the code necessary for preprocessing data, training models, evaluating performance, and saving the best models:

* **depression.py** – Contains code for training and evaluating machine learning models for depression prediction.
* **diabetes.py** – Codebase for diabetes prediction models.
* **heart\_disease.py** – Script that trains and tests models for heart disease prediction.
* **storke.py** – (Possibly a typo for stroke.py) Handles stroke prediction model training and evaluation.

**.53. Performance Plots (.png)**

These images provide visual insights into the performance of different models, including accuracy, AUC scores, and other evaluation metrics:

* **model\_performance\_plot\_depression.png** – A performance visualization for depression prediction models.
* **model\_performance\_plot\_diabetes.png** – A graphical representation of diabetes model performance.
* **model\_performance\_plot\_heart.png** – Performance metrics visualization for heart disease models.
* **model\_performance\_plot\_stroke.png** – A performance graph for stroke prediction models.

**Summary**

This directory effectively organizes both the trained machine learning models and their respective performance evaluations. These assets enable efficient deployment of predictive models while providing meaningful insights into their effectiveness. The Python scripts ensure that the models can be retrained and optimized as necessary, supporting continuous improvement in health risk assessment.