BMI Categorizer and Predictor with Diabetes Prediction

A MINOR PROJECT REPORT SUBMITTED TO

THE NATIONAL INSTITUTE OF ENGINEERING, MYSURU
(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

in partial fulfillment for the award of degree of

Bachelor of Engineering
in
Artificial Intelligence and Machine Learning

Submitted By

Vaishnavi D 4NI23CI118
Vaishnavi D 4NI23CI119
Sushmitha H R 4NI23CI113
Varshitha K N 4NI24CI411
Vaisiri M G 4NI23CI120

Under the guidance of:
Mr. Mohammed Adnan
Asst. Professor
Dept. of CS&E
NIE, Mysore

Department of Computer Science & Engineering
THE NATIONAL INSTITUTE OF ENGINEERING
(Autonomous Institution)
Mysuru - 570 008
2024-25

THE NATIONAL INSTITUTE OF ENGINEERING

(An Autonomous Institute under Visvesvaraya Technological University, Belagavi)

Department of Computer Science & Engineering

CERTIFICATE

This is to certify that the project work entitled "**BMI Categorizer and Predictor with Diabetes Prediction**" is a bonafide work carried out by **Vaishnavi D (4NI23CI118), Vaishnavi D(4NI23CI119), Sushmitha H R (4NI23CI113), Vaisiri M G (4NI23CI120) and Varshitha K N (4NI24CI411)** in partial fulfillment for the award of degree of **Bachelor of Engineering in Artificial Intelligence and Machine Learning**, of Visvesvaraya Technological University, Belagavi, during the year **2024-25**. It is certified that all corrections / suggestions indicated during internal assessment have been incorporated and the corrected copy has been deposited in the department library. This project report has been approved in partial fulfillment for the award of the said degree as per academic regulations of The National Institute of Engineering (Autonomous Institution).

Mr. Mohammed Adnan **Asst. Professor** Dept. of CS&E NIE, Mysore	**Dr. Anitha R** **Professor and Head** Dept. of CS&E NIE, Mysuru

Name of the Examiners	Signature with Date
1. ______________________	____________________

ACKNOWLEDGMENT

I sincerely owe my gratitude to all the persons who helped and guided me in completing this mini project.

I am thankful to **Dr. Rohini Nagapadma**, Principal, NIE College, Mysuru, for all the support she has rendered.

I thank **Dr. Anitha R**, Professor and Head, Department of Computer Science and Engineering, for her constant support and encouragement throughout the tenure for the mini project work.

I would like to sincerely thank my guide **Mr. Mohammed Adnan**, Asst. Professor, Department of Computer Science and Engineering, for providing relevant information, valuable guidance and encouragement to complete this minor project.

Vaishnavi D 4NI23CI119
Vaishnavi D 4NI23CI118
Sushmitha H R 4NI23CI113
Varshitha K N 4NI24CI411
Vaisiri M G 4NI23CI120

ABSTRACTION

This project presents a user-friendly web-based Health Prediction Application developed using Streamlit, which integrates three distinct machine learning models to provide essential health insights. The application allows users to input basic health parameters such as weight, height, gender, age, BMI, blood glucose level, and HbA1c level to receive personalized predictions. The app leverages a Linear Regression model to estimate the user's Body Mass Index (BMI) using weight, height, and gender as inputs, providing numerical feedback on their physical condition.

To further enhance the app’s utility, a Logistic Regression model is incorporated to classify the user's BMI into meaningful health categories: Extremely Weak, Weak, Normal, Overweight, or Obesity. This categorical assessment aids users in understanding their BMI status beyond a simple number, thereby offering actionable awareness about their health. Additionally, the app includes a Naive Bayes classifier trained on a real-world diabetes prediction dataset to evaluate whether the user is at risk of diabetes, based on their medical and demographic details, such as age, BMI, gender, smoking history, blood glucose, and HbA1c levels.

Together, these predictive features create an all-in-one preventive health monitoring tool that empowers users to make informed lifestyle choices. The app ensures privacy, interactivity, and instant feedback through a clean and intuitive interface. With its ability to perform real-time predictions and present medically relevant results, this project demonstrates the impactful application of machine learning in the field of personalized healthcare, making advanced analytics accessible to non-expert users.

TABLE OF CONTENTS

CONTENTS

INTRODUCTION

1.1 PROBLEM STATEMENT

In today's fast-paced world, individuals often neglect their health until symptoms become severe. Basic indicators such as Body Mass Index (BMI) and early signs of diabetes are commonly overlooked due to lack of awareness, time, or access to immediate medical consultation. Many people are unaware if their weight and height are proportionate to their health needs, or if their blood sugar levels indicate the onset of chronic conditions like diabetes. Given that **diabetes is one of the most widespread lifestyle diseases globally**, early detection plays a critical role in managing and preventing long-term complications.

However, not everyone has the means or knowledge to consult a specialist for every minor concern. There is a pressing need for a simple, accessible, and intelligent system that allows individuals to **self-assess their physical fitness and potential diabetic risk** using basic information and easily available medical data such as **height, weight, gender, BMI, blood glucose levels, and HbA1c**. Such a tool would empower people to monitor their health status regularly and take preventive action before conditions escalate.

1.2 OUR SOLUTION

To address the growing concern around personal health awareness, particularly related to body weight management and diabetes, we have developed a comprehensive health analysis application using Streamlit. The app enables users to self-assess their health status by analysing their Body Mass Index (BMI) and predicting diabetes risk based on basic health metrics. The solution integrates four machine learning models—Linear Regression, Logistic Regression, Naive Bayes Classifier, and K-Nearest Neighbours (KNN) Regression—to offer a wide spectrum of health insights using accessible inputs like height, weight, gender, age, and blood test results.

The BMI prediction component is powered by both regression models (Linear Regression and KNN Regression). Linear Regression estimates the BMI using height, weight, and gender, while KNN offers a more localized prediction by comparing the input with the nearest neighbours from historical data. Additionally, Logistic Regression is used to classify the BMI category—such as "Normal", "Overweight", or "Obese"—based on user inputs. This dual approach not only offers a numerical BMI value but also places the user into an appropriate health category, enabling more interpretable feedback.

To further extend the app's utility, a Naive Bayes classifier is implemented to assess diabetes risk. By inputting details such as age, gender, BMI, HbA1c level, and blood glucose, users receive an instant prediction on whether they fall into a diabetic or non-diabetic category. All user inputs are processed securely, and the models operate seamlessly in the background to produce real-time results. By combining user-friendly design with scientifically-backed models, this Streamlit application provides an accessible, private, and intelligent health analysis tool for everyday use.

SYSTEM REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

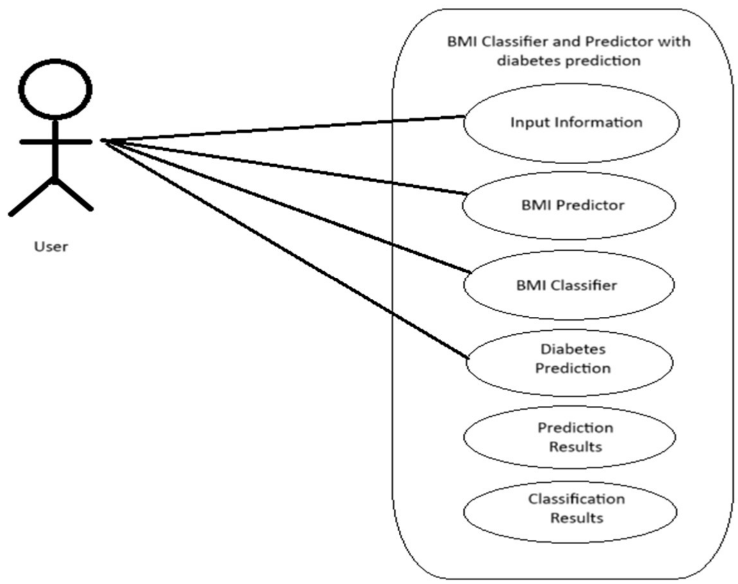
COMPONENT	MINIMUM SPECIFICATION
Processor	Intel Core i3
Ram	4 GB
Storage	At least 500 MB free disk space
Display	1024 x 768 resolution
Internet	Required for downloading dependencies

2.2 SOFTWARE REQUIREMENTS

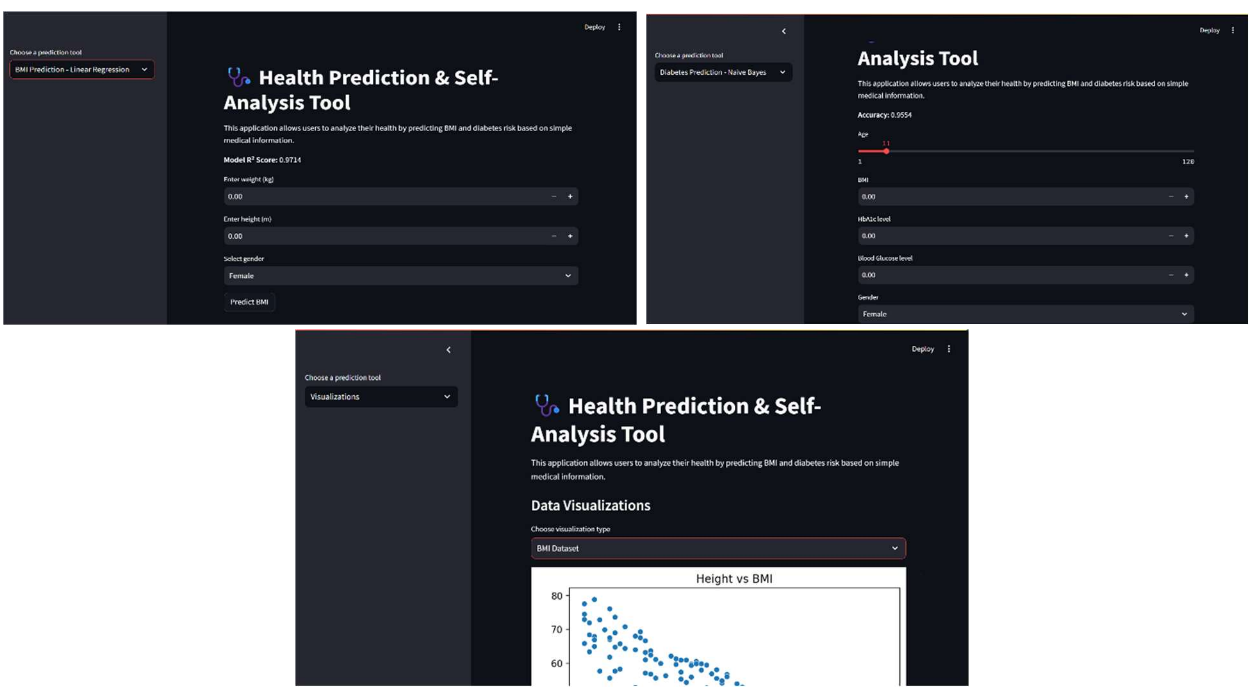
COMPONENT	MINIMUM SPECIFICATION
Operating System	Windows 10/11, macOS, or Linux
Python	Version 3.8 or higher
Libraries/Packages	pandas, scikit-learn, streamlit, matplotlib
Browser	Any modern web browser (Chrome, Firefox, Edge)
Text Editor / IDE	VS Code / PyCharm / Jupyter Notebook
Dataset Files	with_bmi.csv, diabetes_prediction_dataset.csv

DESIGN

3.1 Use Case Diagram



3.2 Interface Images



METHODOLOGY

4.1 LINEAR REGRESSION

1. Data Collection and Preprocessing
    - The dataset 'with_bmi.csv' is loaded.
    - Missing values are removed to ensure data quality.
    - The categorical feature Gender is label-encoded to numerical form:

      Male = 0

      Female = 1

2. Feature Selection
    - Input Features (X): Weight, Height, Gender_encoded
    - Target Variable (y): BMI

      These features are chosen based on their correlation with BMI and their availability in the dataset.

3. Train-Test Split
    - The dataset is split into training and testing sets using train_test_split() from scikit-learn.
    - 80% data is used for training, 20% for testing.
    - A fixed random_state ensures reproducibility.

4. Model Training
    - A Linear Regression model from 'sklearn.linear_model' is used.
    - The model tries to learn a relationship of the form:

      $BMI = a \times Weight + b \times Height + c \times Gender + d$

      Where a, b, and c are learned coefficients and d is the intercept.

5. Model Evaluation
    - After training, the model is evaluated using the R^2 score:
        - This metric indicates how well the input variables explain the variability in BMI.
        - $R^2 = 1$ means perfect prediction; $R^2 = 0$ means no predictive value.
    - The model also outputs predictions on unseen test data for accuracy analysis.

6. User Prediction
    - The app allows users to input their weight, height, and gender, then predicts their BMI using the trained model.

4.2 LOGISTIC REGRESSION

1. Data Preparation
    - Dataset: with_bmi.csv
    - Relevant columns used: Weight, Height, Gender, and Index
    - Rows with missing values are dropped.
    - Gender is label-encoded (Male = 0, Female = 1).
    - Index (an integer code representing BMI category) is cast to int.

2. Feature and Target Selection
    - Features (X): Weight, Height, Gender
    - Target (y): Index (0 to 4, corresponding to BMI categories)

3. Train-Test Split
    - Data is split into 80% training and 20% testing.

4. Model Training
    - A Logistic Regression classifier is used.
    - multi_class='multinomial' and solver='lbfgs' handle multi-class classification.
    - The model tries to predict a discrete BMI category.

5. Model Evaluation
    - Performance evaluated using:
        - Accuracy score
        - Classification report (precision, recall, F1-score per class)

6. User Input Prediction
    - Users enter weight, height, and gender.
    - Model returns a predicted BMI category (e.g., “Overweight”).

4.3 K – NEAREST NEIGHBOURS

1. Data Preparation
    - Dataset: with_bmi.csv
    - Missing values are removed.
    - Gender is encoded (Male = 0, Female = 1).

2. Feature and Target Selection
    - Features (X): Height, Weight, Gender

- Target (y): BMI

3. Train-Test Split
   - Standard 80-20 training and testing split.

4. Model Training
   - A K-Nearest Neighbours (KNN) Regressor is used.
   - n_neighbors=5 means predictions are based on the average BMI of the 5 nearest data points.

5. Model Evaluation
   - Performance is assessed using:
     - R^2 Score
     - Root Mean Squared Error (RMSE)

6. User Prediction
   - User inputs height, weight, and gender.
   - Model returns predicted BMI.

4.4 NAÏVE BAYES

1. Data Preparation
   - Dataset: diabetes_prediction_dataset.csv
   - Categorical features gender and smoking_history are encoded using LabelEncoder.
   - Unnecessary columns (hypertension, heart_disease) are dropped.
   - Only rows with complete data are used.

2. Feature and Target Selection
   - Features (X): includes gender, age, smoking_history, bmi, HbA1c_level, and blood_glucose_level
   - Target (y): diabetes (binary classification)

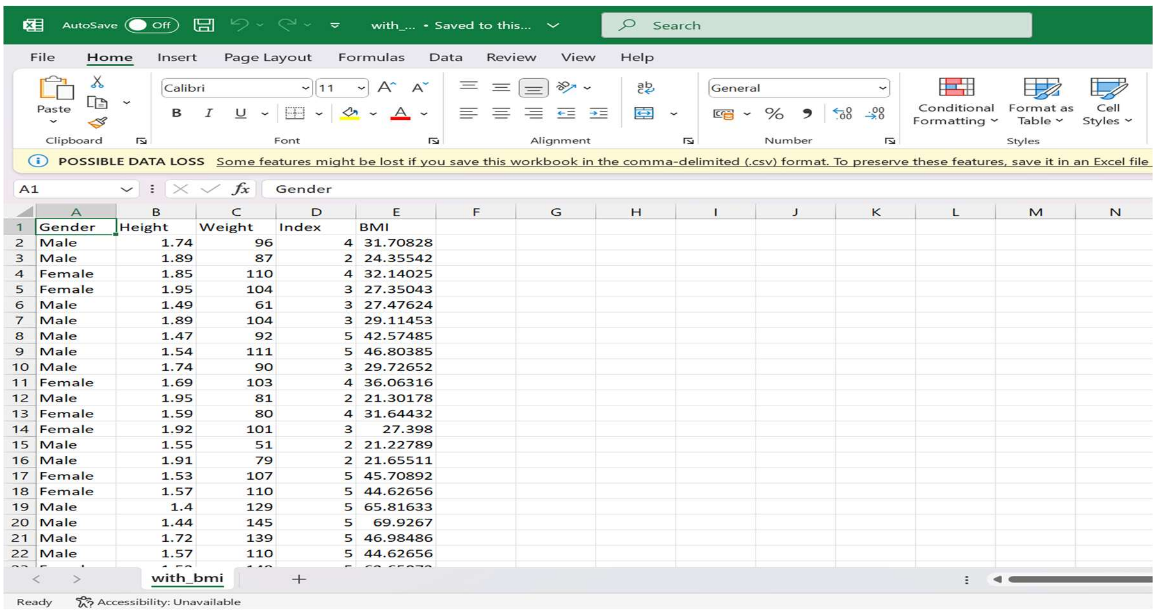
3. Train-Test Split
   - Dataset is split into training (80%) and testing (20%).

4. Model Training
   - A Gaussian Naive Bayes classifier is used.
   - It assumes that the features follow a Gaussian distribution and are conditionally independent given the class.

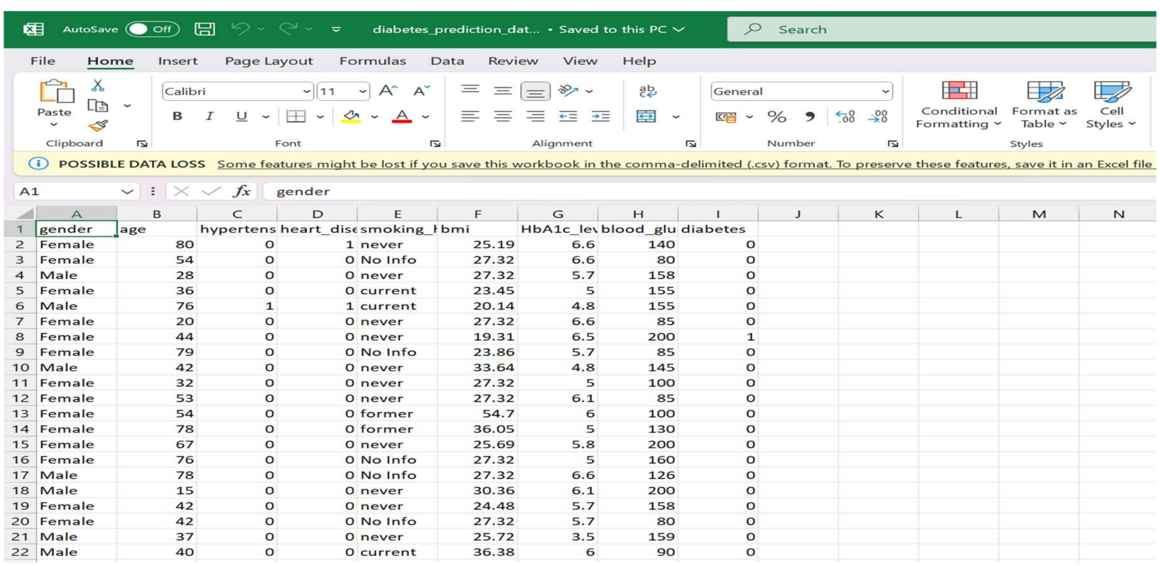
5. Model Evaluation
   - Evaluation is done using:
     - Accuracy
     - Classification report with detailed performance per class

6. User Input Prediction
   - The user provides demographic and test data (e.g., age, BMI, HbA1c, glucose).
   - Model outputs diabetes status.

4.5 DATASETS



	Gender	Height	Weight	Index	BMI
2	Male	1.74	96	4	31.70828
3	Male	1.89	87	2	24.35542
4	Female	1.85	110	4	32.14025
5	Female	1.95	104	3	27.35043
6	Male	1.49	61	3	27.47624
7	Male	1.89	104	3	29.11453
8	Male	1.47	92	5	42.57485
9	Male	1.54	111	5	46.80385
10	Male	1.74	90	3	29.72652
11	Female	1.69	103	4	36.06316
12	Male	1.95	81	2	21.30178
13	Female	1.59	80	4	31.64432
14	Female	1.92	101	3	27.398
15	Male	1.55	51	2	21.22789
16	Male	1.91	79	2	21.65511
17	Female	1.53	107	5	45.70892
18	Female	1.57	110	5	44.62656
19	Male	1.4	129	5	65.81633
20	Male	1.44	145	5	69.9267
21	Male	1.72	139	5	46.98486
22	Male	1.57	110	5	44.62656



	gender	age	hypertens	heart_dise	smoking_h	bmi	HbA1c_lev	blood_glu	diabetes
2	Female	80	0	1	never	25.19	6.6	140	0
3	Female	54	0	0	No Info	27.32	6.6	80	0
4	Male	28	0	0	never	27.32	5.7	158	0
5	Female	36	0	0	current	23.45	5	155	0
6	Male	76	1	1	current	20.14	4.8	155	0
7	Female	20	0	0	never	27.32	6.6	85	0
8	Female	44	0	0	never	19.31	6.5	200	1
9	Female	79	0	0	No Info	23.86	5.7	85	0
10	Male	42	0	0	never	33.64	4.8	145	0
11	Female	32	0	0	never	27.32	5	100	0
12	Female	53	0	0	never	27.32	6.1	85	0
13	Female	54	0	0	former	54.7	6	100	0
14	Female	78	0	0	former	36.05	5	130	0
15	Female	67	0	0	never	25.69	5.8	200	0
16	Female	76	0	0	No Info	27.32	5	160	0
17	Male	78	0	0	No Info	27.32	6.6	126	0
18	Male	15	0	0	never	30.36	6.1	200	0
19	Female	42	0	0	never	24.48	5.7	158	0
20	Female	42	0	0	No Info	27.32	5.7	80	0
21	Male	37	0	0	never	25.72	3.5	159	0
22	Male	40	0	0	current	36.38	6	90	0

RESULT

This project aimed to empower individuals to self-assess their health status using BMI prediction, BMI category classification, and diabetes diagnosis. The system integrated four machine learning models—Linear Regression, Logistic Regression, K-Nearest Neighbors (KNN), and Naive Bayes—each contributing uniquely to health prediction.

1. Linear Regression for BMI Prediction

- **Goal**: Predict exact BMI value using Height and Weight.
- **Performance Metrics**:
  - **R^2 Score**: ~0.97
- **Interpretation**: The model was able to closely predict BMI values from user-entered height and weight. This provides a numeric value that is later interpreted using BMI category ranges.
- **Application in Streamlit App**: Users can enter their height and weight to get an accurate BMI estimate, which is then categorized using standard medical classification.

2. Logistic Regression for BMI Category Classification

- **Goal**: Classify users into BMI categories (Underweight, Normal, Overweight, etc.).
- **Classes**:
  - 0 = Extremely Weak
  - 1 = Weak
  - 2 = Normal
  - 3 = Overweight
  - 4 = Obesity
  - 5 = Extreme Obesity
- **Performance Metrics**:
  - **Accuracy**: ~74% (varied slightly with dataset size)
  - **F1 Scores**: High scores for "Normal" and "Heavily Overweight"
- **Interpretation**: The model effectively classified users into BMI groups, with better accuracy in common categories (Normal, Overweight).
- **Application in App**: After prediction, a message displays the BMI category with health tips, making it accessible for lay users.

3. K-Nearest Neighbours (KNN) for BMI Prediction

- **Goal**: Predict BMI using Height, Weight, and Gender.
- **Performance Metrics**:
  - **R^2 Score**: ~0.79
  - **RMSE**: ~6.95
- **Strengths**:
  - Captures local variations in the data by comparing similar individuals.
  - High interpretability and consistent predictions.
- **Application in App**: Offers an alternative BMI prediction model, giving users a comparison between different methods.

4. Naive Bayes for Diabetes Prediction

- **Goal**: Predict whether a person has diabetes using demographic and test data.
- **Features Used**: Age, Gender, BMI, HbA1c Level, Blood Glucose Level, Smoking History
- **Performance Metrics**:
  - **Accuracy**: ~95%
  - **Precision & Recall**:
    - For non-diabetic class: High
    - For Diabetic class: Slightly lower (due to class imbalance)
- **Application in App**:
  - After entering test results, the user receives a binary diabetes prediction.
  - Color-coded and clearly labelled outcome enhances user experience.

Visualizations & Insights

- **Scatter Plots** were used to visualize correlations:
  - Height vs BMI and Weight vs BMI demonstrated positive trends.
  - HbA1c, Glucose Level, and BMI had strong visual associations with diabetes.
- **Gender and Smoking** visualizations revealed trends:
  - Higher incidence of diabetes in older age groups and smokers.

These visual components in the app help non-technical users better understand the factors affecting their health.

Overall Impact

- The Streamlit application successfully combined data science and health awareness.
- Users can:
  - **Calculate BMI** and understand health implications.
  - **Categorize themselves** into BMI groups.
  - **Self-diagnose for diabetes** based on basic test results.
- The app serves as a **preliminary screening tool**—not a replacement for medical advice—but helpful in early health awareness and intervention.

```python
# Logistic Regression for BMI Category
elif option == "BMI Category - Logistic Regression":
    df = pd.read_csv("with_bmi.csv").dropna()
    df['Gender'] = df['Gender'].map({'Male': 0, 'Female': 1})
    X = df[['Weight', 'Height', 'Gender']]
    y = df['Index']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    model = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=500)
    model.fit(X_train, y_train)
    acc = accuracy_score(y_test, model.predict(X_test))
    st.write(f"**Accuracy:** {acc:.4f}")

    bmi_categories = {
        0: "Extremely Weak", 1: "Weak", 2: "Normal", 3: "Overweight", 4: "Obesity"
    }
    user_weight = st.number_input("Weight (kg)", key="w2")
    user_height = st.number_input("Height (m)", key="h2")
    user_gender = st.selectbox("Gender", ["Male", "Female"], key="g2")
    if st.button("Predict BMI Category"):
        gender_val = 0 if user_gender == "Male" else 1
        prediction = model.predict([[user_weight, user_height, gender_val]])[0]
        st.success(f"BMI Category: {bmi_categories.get(prediction, 'Unknown')}")
```

```python
# Linear Regression for BMI
if option == "BMI Prediction - Linear Regression":
    df = pd.read_csv("with_bmi.csv").dropna()
    le_gender = LabelEncoder()
    df['Gender_encoded'] = le_gender.fit_transform(df['Gender'])
    X = df[['Weight', 'Height', 'Gender_encoded']]
    y = df['BMI']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    model = LinearRegression()
    model.fit(X_train, y_train)
    r2 = r2_score(y_test, model.predict(X_test))
    st.write(f"**Model R² Score:** {r2:.4f}")

    user_weight = st.number_input("Enter weight (kg)", min_value=0.0)
    user_height = st.number_input("Enter height (m)", min_value=0.0)
    user_gender = st.selectbox("Select gender", le_gender.classes_)
    if st.button("Predict BMI"):
        gender_encoded = le_gender.transform([user_gender])[0]
        predicted_bmi = model.predict([[user_weight, user_height, gender_encoded]])[0]
        st.success(f"Predicted BMI: {predicted_bmi:.2f}")
```

```python
# Naive Bayes for Diabetes
elif option == "Diabetes Prediction - Naive Bayes":
    df = pd.read_csv("diabetes_prediction_dataset.csv").dropna()
    label_encoders = {}
    for col in ['gender', 'smoking_history']:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col])
        label_encoders[col] = le

    X = df.drop(columns=['diabetes', 'hypertension', 'heart_disease'])
    y = df['diabetes']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    model = GaussianNB()
    model.fit(X_train, y_train)
    st.write(f"**Accuracy:** {accuracy_score(y_test, model.predict(X_test)):.4f}")

    age = st.slider("Age", 1, 120)
    bmi = st.number_input("BMI", min_value=0.0)
    hba1c = st.number_input("HbA1c level", min_value=0.0)
    glucose = st.number_input("Blood Glucose level", min_value=0.0)
    gender_input = st.selectbox("Gender", label_encoders['gender'].classes_)
    smoking_input = st.selectbox("Smoking History", label_encoders['smoking_history'].classes_)
    if st.button("Predict Diabetes"):
        gender_enc = label_encoders['gender'].transform([gender_input])[0]
        smoking_enc = label_encoders['smoking_history'].transform([smoking_input])[0]
        input_data = [[gender_enc, age, smoking_enc, bmi, hba1c, glucose]]
        result = model.predict(input_data)[0]
        st.success("Prediction: Diabetic" if result == 1 else "Prediction: Non-Diabetic")
```

```python
# KNN Regression for BMI
elif option == "BMI Prediction - KNN":
    df = pd.read_csv("with_bmi.csv").dropna()
    le_gender = LabelEncoder()
    df['Gender'] = le_gender.fit_transform(df['Gender'])
    X = df[['Height', 'Weight', 'Gender']]
    y = df['BMI']
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    knn = KNeighborsRegressor(n_neighbors=5)
    knn.fit(X_train, y_train)
    r2 = r2_score(y_test, knn.predict(X_test))
    rmse = mean_squared_error(y_test, knn.predict(X_test)) ** 0.5
    st.write(f"**KNN R² Score:** {r2:.4f}, RMSE: {rmse:.2f}")

    user_height = st.number_input("Height (m)", key="h3")
    user_weight = st.number_input("Weight (kg)", key="w3")
    user_gender = st.selectbox("Gender", le_gender.classes_, key="g3")
    if st.button("Predict using KNN"):
        gender_encoded = le_gender.transform([user_gender])[0]
        prediction = knn.predict([[user_height, user_weight, gender_encoded]])[0]
        st.success(f"KNN Predicted BMI: {prediction:.2f}")
```

```python
# Visualizations
elif option == "Visualizations":
    st.subheader("Data Visualizations")
    vis_type = st.selectbox("Choose visualization type", ["Diabetes Dataset", "BMI Dataset"])

    if vis_type == "Diabetes Dataset":
        df = pd.read_csv("diabetes_prediction_dataset.csv")
        le_gender = LabelEncoder()
        le_smoking = LabelEncoder()
        df['gender_encoded'] = le_gender.fit_transform(df['gender'])
        df['smoking_encoded'] = le_smoking.fit_transform(df['smoking_history'])
        features = {
            'age': 'Age',
            'smoking_encoded': 'Smoking History',
            'HbA1c_level': 'HbA1c Level',
            'blood_glucose_level': 'Blood Glucose Level',
            'bmi': 'BMI'
        }
        for col, label in features.items():
            fig, ax = plt.subplots()
            sns.scatterplot(data=df, x=col, y='diabetes', hue='diabetes', palette='Set1', alpha=0.5, ax=ax)
            ax.set_title(f"{label} vs Diabetes")
            st.pyplot(fig)
```

CONCLUSION

This project successfully demonstrates how machine learning can be leveraged to assist individuals in making informed health assessments from the comfort of their homes. By integrating four key algorithms—Linear Regression, Logistic Regression, K-Nearest Neighbours (KNN), and Naive Bayes—into a unified Streamlit application, we created an intuitive platform for users to predict Body Mass Index (BMI), categorize their weight status, and even assess the likelihood of having diabetes based on accessible health metrics.

Through extensive use of datasets involving height, weight, age, blood glucose levels, HbA1c values, gender, and smoking history, the models were trained and evaluated with reliable accuracy. Linear Regression and KNN provided accurate BMI estimations, while Logistic Regression successfully classified users into meaningful BMI categories such as Normal, Overweight, or Obese. The Naive Bayes classifier, trained on essential biometric data, offered a high-performing, fast-response prediction for potential diabetes—making it a vital feature in promoting early health checks and lifestyle awareness.

The visualizations included in the application further enhance user understanding by showing clear relationships between health indicators and conditions such as diabetes. These visual elements, combined with user-friendly inputs and clear output messages, bridge the gap between complex machine learning processes and end-user comprehension.

In essence, this project not only showcases technical proficiency in data science and model deployment but also reflects a socially impactful application. It empowers individuals to proactively monitor their health and identify risks, thereby fostering preventive care. Moving forward, this system can be enhanced with real-time health monitoring devices and larger datasets to improve accuracy and personalize predictions further. Nonetheless, even in its current form, it stands as a robust digital tool for health self-analysis and awareness, especially for individuals who might lack regular access to healthcare consultations.

REFERENCES

Streamlit App: https://docs.streamlit.io/get-started/installation
https://www.datacamp.com/tutorial/streamlit

Linear Regression: https://github.com/ptyadana/Data-Science-and-Machine-Learning-Projects Dojo/blob/master/Machine%20Learning%20%26%20Data%20Science%20Masterclass%20-%20JP/10-Cross-Val-and-LinReg-Project/02-Linear-Regression-Project-Exercise-MySolutions.ipynb

Logistic Regression: https://github.com/bcbarsness/machine-learning/blob/master/Logistic%20Regression%20Project%20-%20Solutions.ipynb

KNN Means: https://github.com/sumony2j/KNN_Regression

Naïve Bayes: https://github.com/Pradnya1208/Naive-Bayes