**Diabetes Prediction Project**

**Introduction**:

The **Diabetes prediction dataset** is a collection of medical and demographic data from patients, along with their diabetes status (positive or negative). The data includes features such as age, gender, body mass index (BMI), hypertension, heart disease, smoking history, HbA1c level, and blood glucose level. This dataset can be used to build machine learning models to predict diabetes in patients based on their medical history and demographic information. This can be useful for healthcare professionals in identifying patients who may be at risk of developing diabetes and in developing personalized treatment plans. Additionally, the dataset can be used by researchers to explore the relationships between various medical and demographic factors and the likelihood of developing diabetes. **Promoting early detection, personalized treatment, and lifestyle modifications through data-driven insights reduces diabetes risks, enhances patient care, and improves healthcare efficiency.**

**Link dataset**: [Diabetes prediction dataset](https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset/data)   
   
**Related Work:**   
   
Link to the previous project: [✅ Diabetes Prediction 💊](https://www.kaggle.com/code/zabihullah18/diabetes-prediction)

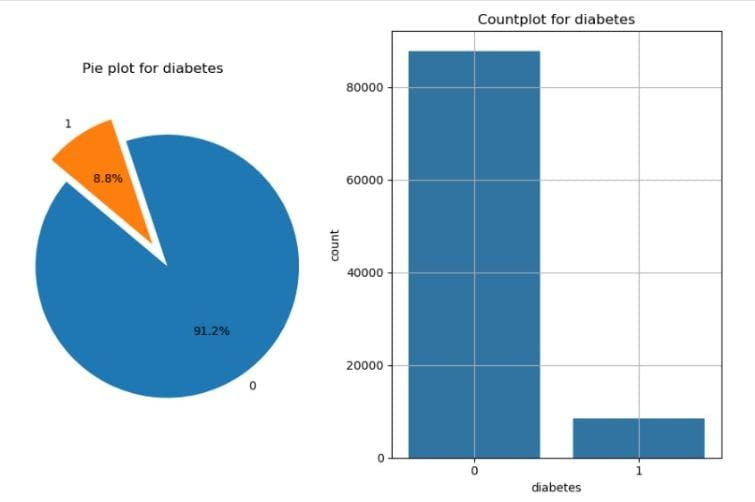
Methodology:

1- The code imports essential Python libraries: **pandas** for data manipulation, **numpy** for numerical computations, **matplotlib.pyplot** for visualizations, and **seaborn** for advanced, aesthetically pleasing data visualizations. 

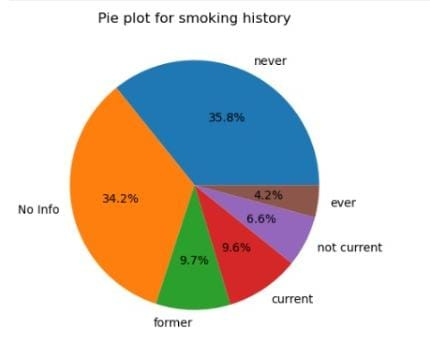
2- In the preprocessing: The dataset initially contained **100,000 rows and 9 columns**. After identifying **3,854 duplicated rows** using data.duplicated().sum(), they were removed with data.drop\_duplicates(inplace=True), reducing the data to **96,146 rows**. Missing values were then checked using data.isna().sum(). The data.info() and data.describe() commands provided insights into the dataset's structure and statistical summary. Rows where the gender column contained "Other" were dropped (data.drop()), ensuring only "Male" and "Female" remain, resulting in **96,128 rows**. Finally, the unique categories in the gender and smoking history columns were confirmed using .unique(), revealing valid values for further analysis. 

3-visualization:

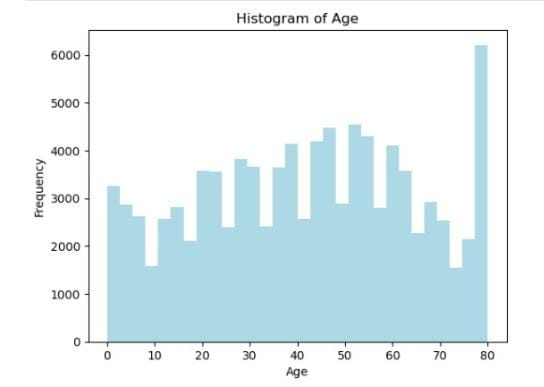
* **Pie Chart:** Displays the percentage distribution of diabetes cases (positive and negative).
* **Count Plot:** Shows the absolute count of each category.
* This analysis provides a quick visual understanding of the overall distribution of diabetes-related



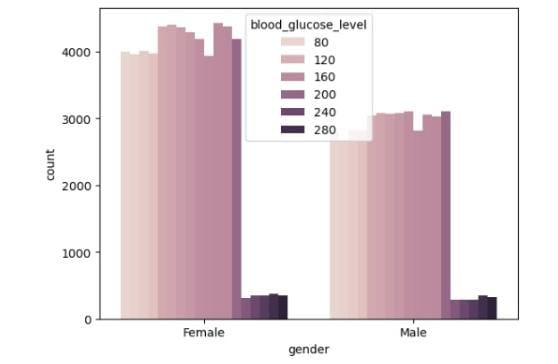
* The pie chart will display the proportion of different smoking history categories, providing a visual representation of the smoking status in the dataset.



* The histogram will show the frequency distribution of the **age** values in the dataset, providing insights into the age range and concentration of data points in various age groups



* The histogram will show the frequency distribution of the **age** values in the dataset, providing insights into the age range and concentration of data points in various age groups



4- Feature selection: The target variable diabetes was separated from the dataset, with features stored in X and labels in y. The data was split into **training (80%)** and **testing (20%)** sets using train\_test\_split. To normalize the feature values, **MinMaxScaler** was applied, scaling X\_train and X\_test to a range between 0 and 1. This ensures consistent feature scaling, improving model performance.

5- Models

1. The **Random Forest Classifier** achieved an accuracy of **95%**. Class 0 showed excellent performance with **precision 0.97** and **recall 0.97**, while class 1 had moderate performance with **precision 0.74** and **recall 0.75**. The model performs well overall but slightly underperforms for the minority class.
2. The **Gradient Boosting Classifier** achieved an accuracy of **95%**. Class 0 performed strongly with **precision 0.97** and **recall 0.97**, while class 1 showed moderate performance with **precision 0.74** and **recall 0.75**. Overall, the model is effective but could improve on the minority class.
3. The **SVC model with a linear kernel** achieved an overall accuracy of **95.6%**. Class 0 performed well with **precision 0.96** and **recall 0.99**, while class 1 had lower recall (**0.57**) but decent precision (**0.90**). The model favors the majority class.
4. The **K-Nearest Neighbors (KNN)** model achieved an accuracy of **90.8%**. Class 0 showed excellent performance with **precision 0.98**, while class 1 had lower precision (**0.49**) but higher recall (**0.78**). The model favors the majority class.

Link Colab of project: [Google Colab](https://colab.research.google.com/drive/18QERLDZrTZSaTNouhDchsZVW_RZu3o4j?usp=sharing) 

Discussion:    
   
 Project of Related Work   
**Evaluation of the Random Forest Model**

* **Training Accuracy**: 89.2%
* **Validation Accuracy**: 87.6%

74b6efd8-010b-464e-9d82-d75cbf2a5112.png

**Our project**

**Evaluation of the Random Forest Model**

* Training Accuracy: 96%
* Validation Accuracy: 95%