***Diabetes Dataset***

**Team Members & Collaboration:**

Yasmeen Abuhaltem: Data cleaning, model fitting, data splitting prediction and evaluation. Description of steps.

Hanan Njoum: Data scaling, Data normalization code. Flowchart, report editing.

Dana Ayyad: Data cleaning, decision tree approach (corona dataset), model evaluation, scaling and normalization tables & code.

**Introduction:**

In this project, our team will be participating in Kaggle competitions to tackle challenging machine learning tasks. In this project we will be focusing on two datasets: the "Diabetes Prediction Dataset" and the "Corona Virus Latest Data 2023." We will explore these datasets, apply various approaches, compare the results, and document our findings in this report.

The Diabetes Prediction Dataset presents a challenge in the healthcare domain of predicting the incidence of diabetes based on various health indicators. Meanwhile, the dataset "Coronavirus Latest Data 2023" plunges into the world of pandemics. Looking at the latest data on COVID-19 cases, deaths and recoveries will help us understand the dynamics of the virus and its impact on different locations.

**About the dataset:**

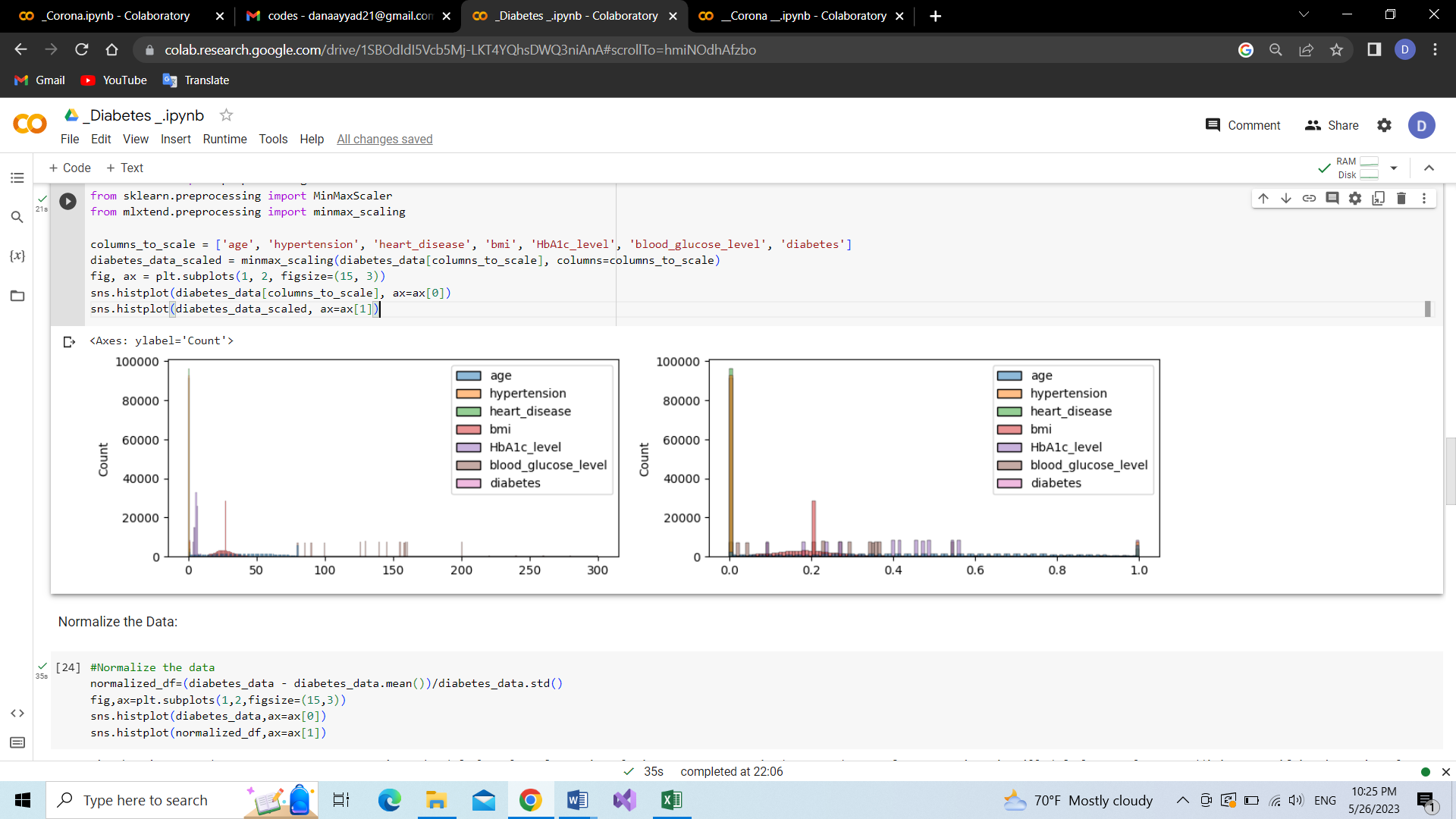
In the provided dataset, which can be found at the following link: https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset, the focus is on predicting diabetes diagnosis using various health-related features. It is a classification problem where we aim to build a model that can classify new or unknown cases into two classes: "diabetic" and "non-diabetic".

The dataset contains several demographic and health-related attributes for each individual, including age, body mass index (BMI), blood pressure, insulin level, and glucose level, among others. These attributes serve as input features that can be used to predict the target variable, which is the diabetes diagnosis.

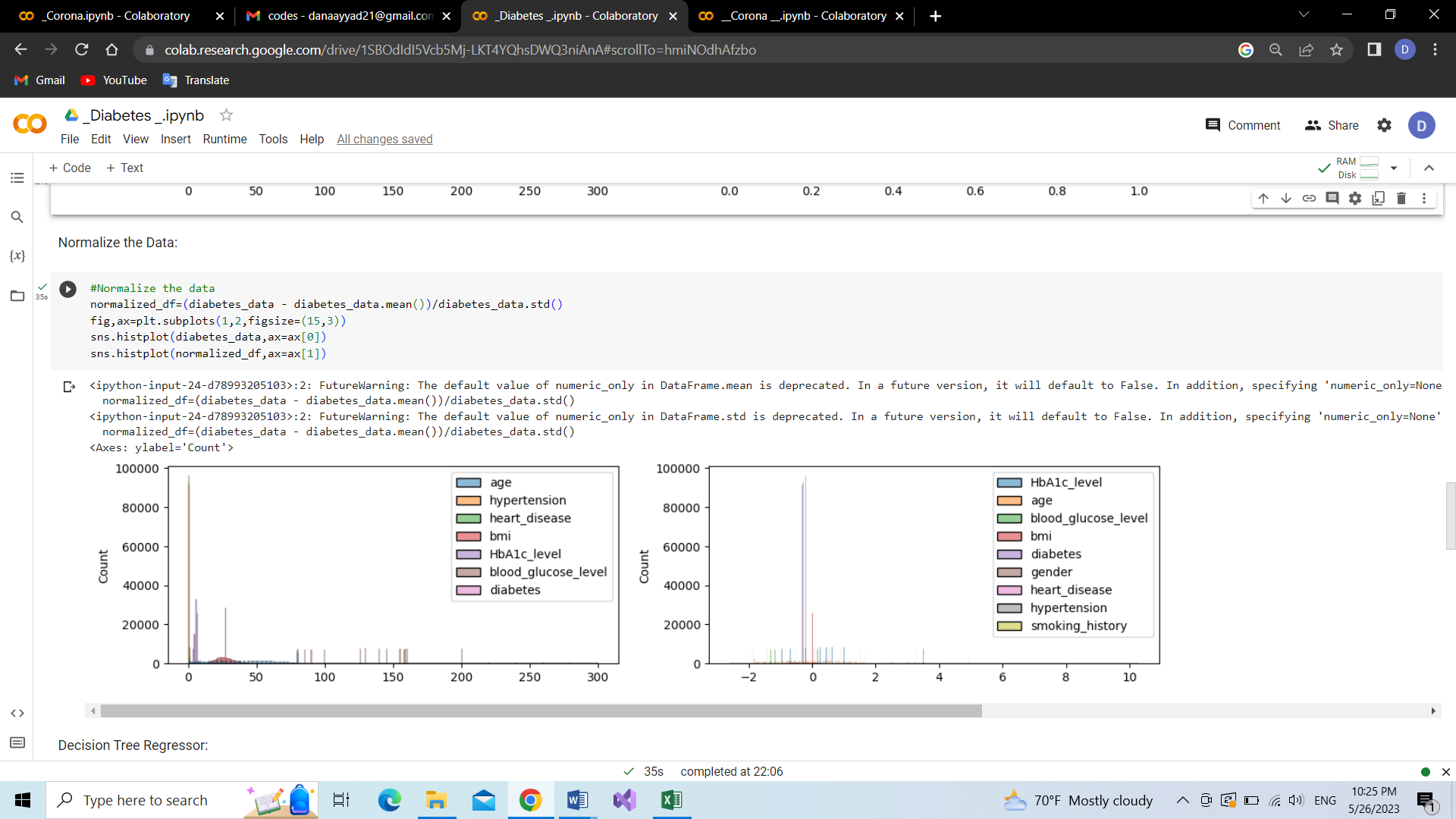
**Description of the Methodologies and Steps:**

1. Loading the dataset: The dataset, "diabetes\_prediction\_dataset.csv," is loaded using the pandas library (imported from google drive). The dataset contains information related to diabetes prediction.
2. Data exploration and preprocessing:

* Missing value analysis: The missing\_values\_count variable calculates the number of missing values in each column. The percentage of missing values is then calculated and printed.
* Data scaling: The minmax\_scaling function from the mlxtend library is used to scale the data between 0 and 1, ensuring consistency.



* Data normalization: The Box-Cox transformation is applied to normalize the data, reducing skewness, and improving model performance.



* Handling missing values: Missing values represented as "No Info" are replaced with NaN values. The bfill method is used to fill missing values with the previous non-null value in each column. Finally, any remaining missing values are filled with 0.
* Correcting incorrect values: Incorrect values in the 'age' column, such as values less than 1 or non-integer values, are replaced with the value 1.

1. Model fitting and prediction:

* Decision Tree Regression: The dataset is split into features (X) and the target variable (y). Categorical features are one-hot encoded. A decision tree regression model is then created and fitted to the data.
* Mean Absolute Error (MAE): The mean absolute error is calculated by comparing the actual target values with the predicted values.

1. Train-Test Split and Model Evaluation:

* Train-Test Split: The dataset is split into training and validation sets using the train\_test\_split function from scikit-learn.
* Decision Tree Regression with MAE: A decision tree regression model is trained using the training set and evaluated using the validation set by calculating the mean absolute error.
* Varying Maximum Leaf Nodes: The mean absolute error is calculated for different values of maximum leaf nodes in the decision tree model, including 5, 50, 250, 500, and 5000, to find the optimal number of leaf nodes.

1. Random Forest Regression:

* Random Forest Regression: A random forest regression model is created using the RandomForestRegressor from scikit-learn and trained on the training set.
* MAE Calculation: The mean absolute error is calculated by comparing the predicted values with the actual values using the validation set.

1. Methodologies Used:

* Decision Tree Regression
* Random Forest Regression

Start

Dataset is split into training and validation sets

Create and fit decision tree regression model

Load the dataset

model

Predict values

Calculate mean absolute error

Create and fit random forest regression model

Change values of maximum leaf nodes

Predict values

Dataset is split into features (X) and target variable (y)

End

Model is trained using the training set and evaluated using the validation set

Model is trained using the training set and evaluated using the validation set

Calculate mean absolute error

Data Scaling

Data Normalization

Missing values analysis

Handling missing values

Correcting incorrect values

**Conclusion:**

Both Decision Tree Regression and Random Forest Regression were employed to predict diabetes in the provided code. The choice of the better model depends on various factors, including performance metrics, interpretability, and the specific characteristics of the dataset.

In this dataset the methodology with better performance is the method that results in lower MAE(mean absolute error). The MAE for the Decision Tree Regression is 0.048968504495504495 and the MAE for the Random Forest Regression is 0.048924643492533476.Therefore the best methodology for this dataset is the Random Forest Regression Methodology.