**Data Engineering Project**

This project aims to explore and analyze stroke-related health data to identify potential risk factors and patterns. The key goal is to understand how different health and lifestyle features—such as age, glucose level, BMI, and smoking status—relate to the likelihood of having a stroke. By using data analysis and visualization techniques, we aim to uncover meaningful insights that could support early detection and prevention strategies. This work is important because stroke is a leading cause of death and disability worldwide, and data-driven analysis can help improve public health awareness and decision-making.

**Introduction**

Stroke is a serious medical condition that occurs when the blood supply to the brain is interrupted, leading to brain damage and potentially long-term complications. Understanding what factors contribute to stroke risk is crucial for both prevention and treatment. This project focuses on analyzing a real-world healthcare dataset to study how factors like age, body weight, blood sugar, and lifestyle choices (like smoking or work type) relate to stroke.

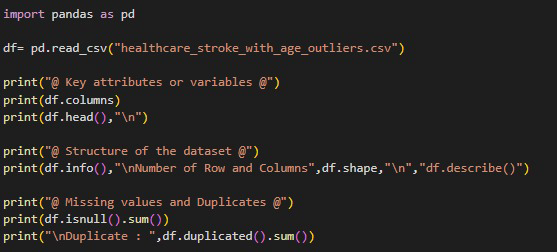
**Key Features Explained:**

Here are short and simple explanations of the main columns in the dataset:

* **age**: The age of the person. Age is a critical factor in stroke risk.
* **gender**: Whether the person is male, female, or other.
* **hypertension**: 1 if the person has high blood pressure, 0 if not.
* **heart\_disease:** 1 if the person has heart disease, 0 if not.
* **ever\_married:** Indicates if the person has ever been married.
* **work\_type:** The type of job (Private, Self-employed, Government job.).
* **Residence\_type:** Whether the person lives in a rural or urban area.
* **avg\_glucose\_level:** The average blood sugar level of the person. High levels may indicate diabetes or risk of stroke.
* **bmi (Body Mass Index):** A number that shows if someone is underweight, normal, overweight, or obese. It’s calculated from height and weight.
* **smoking\_status:** Whether the person smokes, smoked before, or never smoked.
* **stroke**: 1 if the person has had a stroke, 0 if not. This is the target variable we are trying to understand.

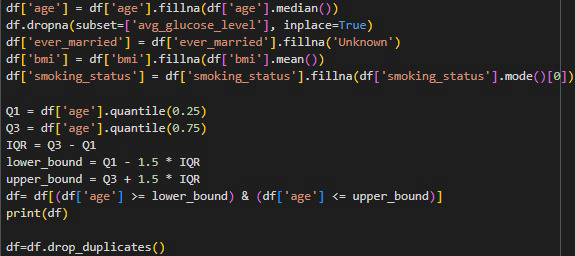
1. **Dataset Description and Challenges**

reading the dataset from a CSV file named \*"healthcare\\_stroke\\_with\\_age\\_outliers.csv"\*`. The code then displays the column names and the first few rows to give an overview of the data. It proceeds to examine the structure of the dataset using `info()` and `shape`, which reveal the data types, non-null counts, and the number of rows and columns. Although the line `print("df.describe()")` is written as a string and does not execute, it is intended to display summary statistics of numerical columns. Finally, the code checks for missing values in each column and counts any duplicate rows in the dataset, helping to identify potential data quality issues.

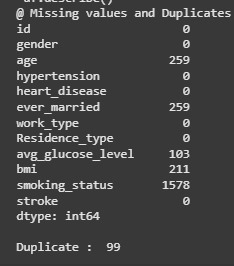
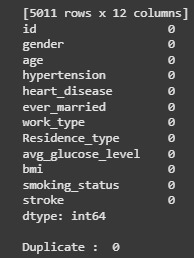


1. **Data Engineering (Data Preprocessing)**

**-Cleaning:**

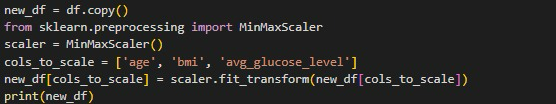
This code performs data cleaning by handling missing values, removing outliers, and eliminating duplicates. First, it fills missing values in the 'age' column with the median age, and drops rows with missing 'avg\_glucose\_level' values. The 'ever\_married' column is filled with the placeholder 'Unknown', missing 'bmi' values are replaced with the mean BMI, and the most frequent category (mode) is used to fill missing values in the 'smoking\_status' column. After addressing missing data, the code identifies and removes outliers in the 'age' column using the Interquartile Range (IQR) method—keeping only those within the calculated lower and upper bounds. Finally, the code removes any duplicate rows to ensure the dataset contains only unique records, making it clean and ready for analysis.

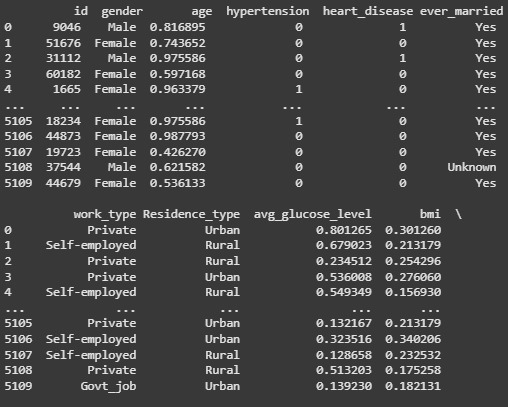
**Output:**



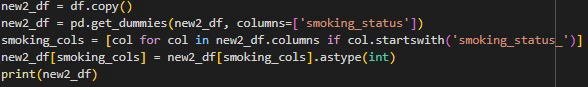
**-Transformation**:

**Min-Max Scaling:** This code creates a copy of the cleaned dataset and applies feature scaling to selected numerical columns. It begins by importing the MinMaxScaler from sklearn.preprocessing and initializing it. The columns chosen for scaling are 'age', 'bmi', and 'avg\_glucose\_level', which are then transformed so that their values fall within a fixed range of 0 to 1. This normalization ensures that these features are on the same scale, which can improve the performance of machine learning models that are sensitive to feature magnitude.

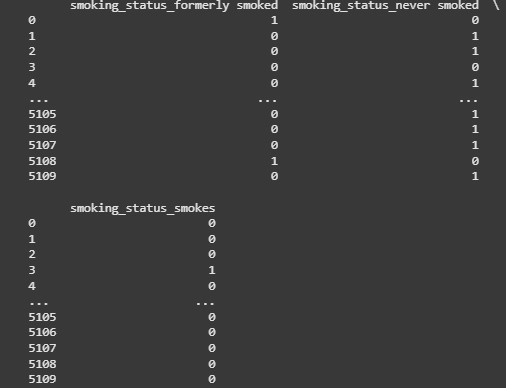


**Output:**

**One-Hot Encoding:** This code applies \*\*one-hot encoding\*\* to the \*\*'smoking\\_status'\*\* column in the `new2\_df` DataFrame using `pd.get\_dummies()`, which creates separate binary columns for each category in the original column. It then collects the names of all new columns that start with `'smoking\_status\_'` into a list called `smoking\_cols`. After that, it ensures that the values in these new columns are explicitly converted to integers (0 or 1) using `astype(int)`. This is useful for maintaining data type consistency and ensuring compatibility with machine learning models. \*\*One-hot encoding\*\* was chosen over \*\*label encoding\*\* because the \*\*'smoking\\_status'\*\* categories are nominal (no natural order). Using label encoding would assign arbitrary numeric values that could mislead models into assuming a ranked relationship between categories, while one-hot encoding treats each category independently and avoids introducing unintended bias.

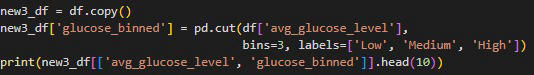


Output:

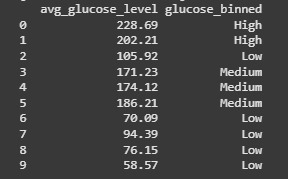


**Fixed-Width Binning:** This code creates a new categorical column called 'glucose\_binned' in the new3\_df DataFrame by dividing the 'avg\_glucose\_level' values into three equal-width bins using pd.cut().

The bins are labeled as 'Low', 'Medium', and 'High', which helps group continuous glucose values into meaningful categories. This transformation is useful for simplifying analysis or preparing the data for models that handle categorical features better than continuous ones.



Output:



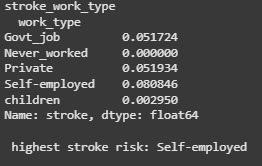
**3-Data Analysis:**

In these codes, three key pandas functions—groupby, filter, and agg—were used to explore and analyze the dataset. The groupby function was used to organize the data into groups based on one or more categorical features, such as 'work\_type', 'Residence\_type', or combinations like 'gender' and 'Residence\_type'. This allows us to analyze patterns within each group.

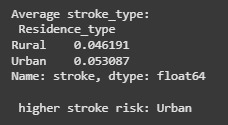
The agg() function was then used after groupby to apply aggregation methods like calculating the mean or sum for specific columns (e.g., stroke rate or average glucose level), helping to summarize the data within each group.

Meanwhile, filtering was used to extract specific subsets of the data based on conditions, such as patients with glucose levels over 200, those with both hypertension and heart disease, or smokers with heart disease. Filtering helps isolate and analyze specific target groups in the dataset for more focused insights.

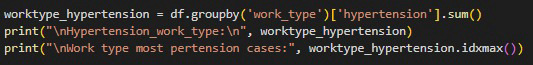
**-Group the data by 'work\_type' and compute the average stroke rate. Which work category has the highest risk of stroke?**

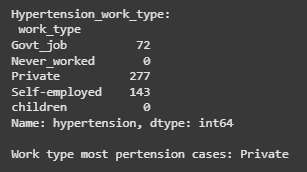
**Output:** 

**-Group the data by ‘Residence\_type’ and compute the average stroke rate. Which residence type has a higher stroke risk?** 

**Output:** 

**-Group the data by ‘work\_type’ and count the number of people with hypertension. Which work type has the highest number of hypertension cases?**



**Output:** 

**-Filter the data to find how many patients have an average glucose level greater than 200.**



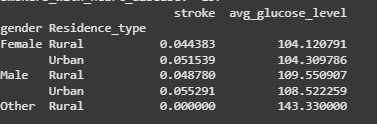
**-Filter the data to find how many patients have both hypertension and heart disease.** 

**Output:**

**-Filter the data to find how many smokers (current or former) have heart disease.** 



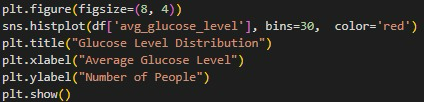
**-Group the data by both ‘gender’ and ‘Residence\_type’, and compute the average stroke rate and average glucose level for each group.** 

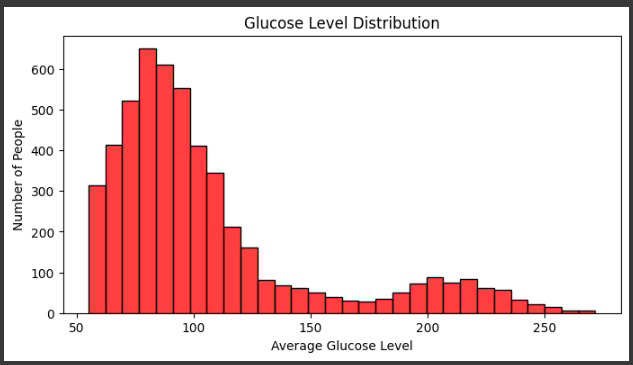
**Output:** 

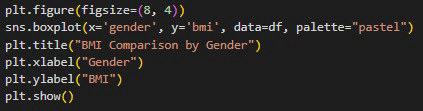
In this analysis, groupby, agg, and filtering techniques were used to summarize and explore the data. groupby helped organize the data into categories, while agg calculated useful statistics like averages. filtering allowed us to focus on specific conditions or groups, such as high glucose patients or smokers with heart disease. These techniques made it easier to extract meaningful insights and identify patterns in the dataset.

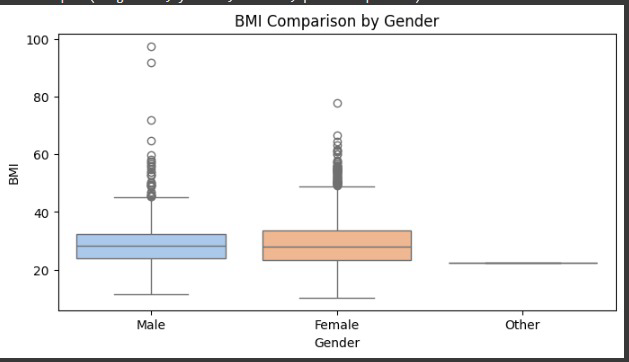
## **4-Data Visualization:**

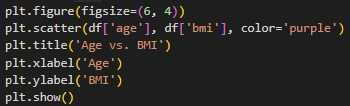
1. **Histogram – Glucose Level Distribution:** Shows how average glucose levels are spread among people. Most values are between 80–150, with some high outliers. This helps spot people at risk of diabetes or stroke.

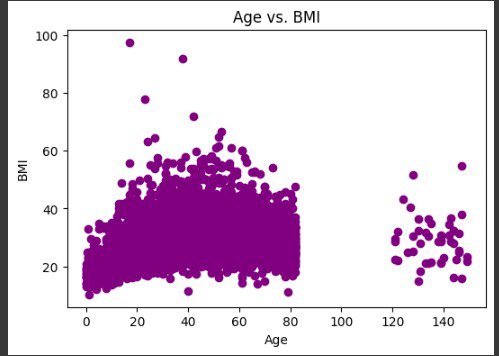


**Output:** 

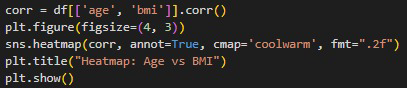
**2. Box Plot – BMI by Gender:**Compares BMI values for males and females. It shows the median and range for each group. This helps us see if one gender tends to have a higher BMI.

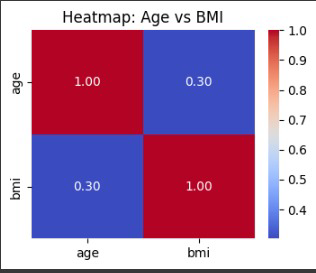
Output:

1. **Scatter Plot – Age vs. BMI:** Plots age against BMI. There is no clear pattern, meaning age and BMI are not strongly related in this dataset.

 Output:

1. **Heatmap – Correlation Between Age and BMI:** Confirms the weak correlation between age and BMI with a low correlation value close to 0.



Output:

**Insights**

These visualizations helped highlight key patterns, like common glucose levels, BMI differences by gender, and the lack of a strong link between age and BMI. This supports the goal of the project: to understand health factors that may relate to stroke risk.

**Conclusion**

This project analyzed a healthcare dataset to understand the key factors associated with stroke. Through data cleaning, transformation, and visualization, we found that certain features—such as age, high glucose levels, hypertension, and smoking—are closely related to stroke risk. For example, people with high glucose levels (above 200) and those who smoke or have a history of smoking showed a higher incidence of heart disease and stroke. Also, work type and residence location showed some differences in average stroke rates.

Name: Khalid Ghanayem

ID: 202211341

GitHub Link: [Data Engineering Project](https://github.com/Khalid0415/Data-Engineering-Project)