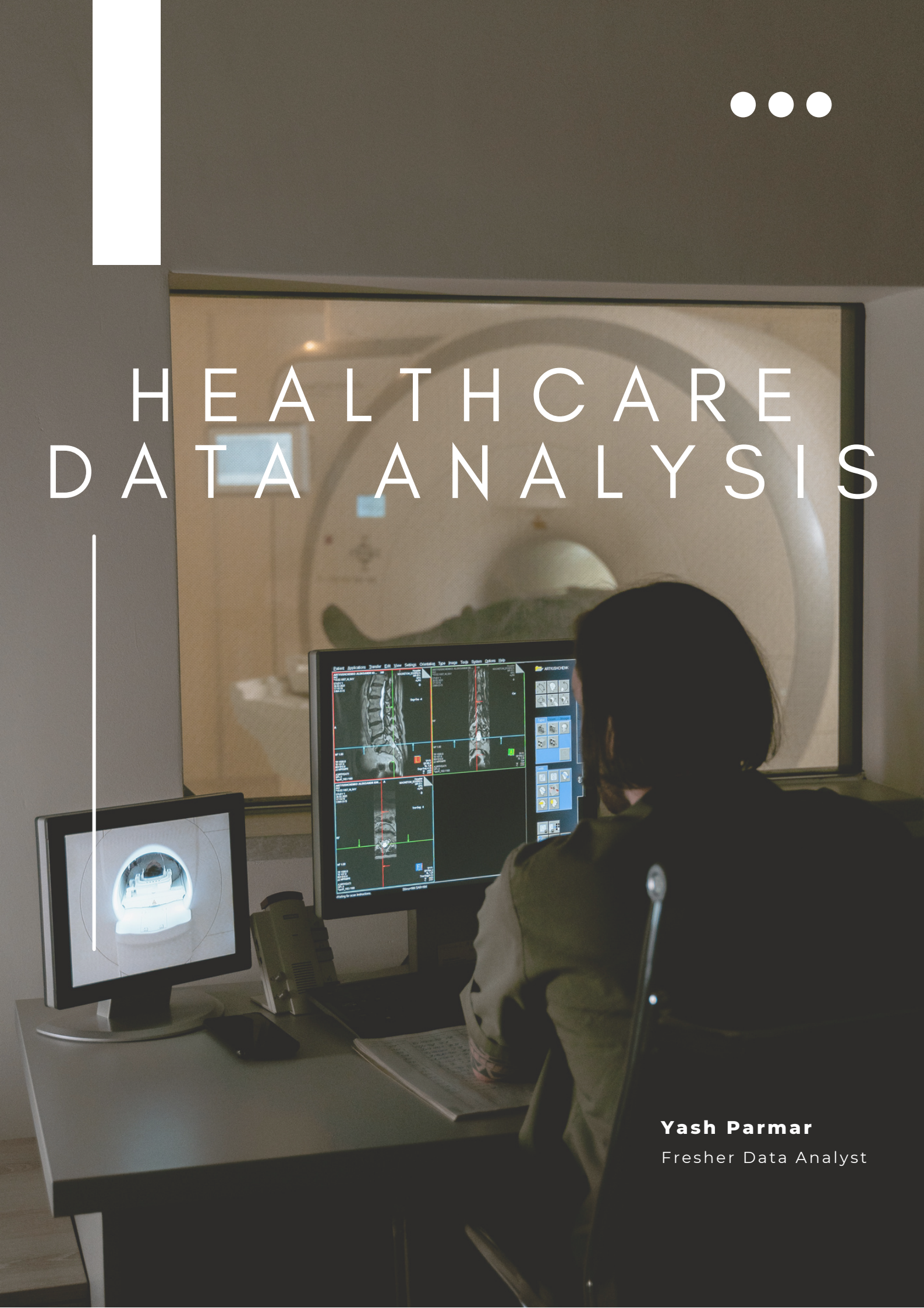
HEALTHCARE DATA ANALYSIS



Yash Parmar
Fresher Data Analyst

1. Introduction:

Welcome to the Healthcare Data Analysis and Machine Learning Project Report. The primary goal of this project is to explore a dummy healthcare dataset, perform data analysis, implement data preprocessing techniques, feature engineering, and ultimately build a machine learning model for disease classification. This report presents a comprehensive overview of the entire process, from data generation to model evaluation.

Dataset Used:

The dataset employed in this project was synthetically created to mimic healthcare-related data. It consists of various health parameters such as fever, blood pressure, cholesterol level, and more. The dataset has been divided into two parts, each simulating different aspects of medical data.

Main Steps Covered:

This report is structured to cover the following main steps:

- **Data Generation**: A detailed explanation of how the dummy datasets were created using Python's Pandas and NumPy libraries.
- **Data Analysis:** An exploration of the dataset's statistical properties, including summary statistics and null value analysis.
- Handling Missing Values: Discussion on how missing values were treated for both numerical and categorical features.
- Exploratory Data Visualization: Presentation of histograms and box plots to visualize data distributions and potential outliers.
- Data Transformation: Explanation of the one-hot encoding for gender, age binning, and the creation of a new blood pressure category feature.
- Feature Engineering: Describing the process of generating meaningful features for disease classification.
- Machine Learning Model: Explanation of the Gradient Boosting Classifier used for disease prediction and the evaluation of model accuracy.
- Classification Report: Presentation of precision, recall, F1-score, and support metrics to assess the model's performance.

Through this report, we will gain insights into the steps taken to understand, clean, transform, and analyze healthcare data, and how a machine learning model can be applied to predict diseases based on the provided features.

2. Data Preparation:

Generating Dummy Datasets:

In my project, I have generated two dummy datasets using Python's random and dumpy libraries. These datasets simulate health-related information for patients. I have created two separate datasets named healthcare_dataset.csv and healthcare_dataset2.csv.

Dataset Structure and Columns: Both datasets have similar structures with the following columns:

- 'Diseases' (or 'Problem'): This column represents the health condition or disease that the patient has. It contains various diseases such as "Influenza," "Diabetes," etc.
- 'Fever': Indicates the patient's body temperature in Fahrenheit. It's a numerical value or NaN (missing).
- 'Cough': Represents whether the patient has a cough (True), doesn't have a cough (False), or the data is missing (NaN).
- 'Fatigue': Indicates whether the patient is experiencing fatigue (True), not experiencing fatigue (False), or the data is missing (NaN).
- 'Difficulty Breathing': Indicates whether the patient has difficulty breathing (True), doesn't have difficulty breathing (False), or the data is missing (NaN).
- 'Age': Represents the age of the patient. It's a numerical value or NaN (missing).
- 'Gender': Represents the gender of the patient. It can be "Male," "Female," or the data is missing (NaN).
- 'Systolic Blood Pressure' (SBP): Represents the systolic blood pressure of the patient. It's a numerical value or NaN (missing).
- 'Diastolic Blood Pressure' (DBP): Represents the diastolic blood pressure of the patient. It's a numerical value or NaN (missing).
- 'Cholesterol Level': Indicates the cholesterol level of the patient. It's a numerical value or NaN (missing).

Assumptions and Issues: In generating the datasets, a few assumptions and issues can be noted:

- The diseases and problems are randomly assigned from a predefined list, which might lead to duplicates or non-realistic combinations.
- The temperature, blood pressure, and cholesterol values are generated within specific ranges, which may not accurately represent real-world values.
- The use of randomization to create missing values might not mimic the patterns of real missing data in actual health records.
- The datasets are relatively small (500 and 700 samples), which might impact the performance of machine learning models due to limited diversity.

Overall, the generated datasets serve as a simplified representation for learning the data analysis and machine learning process, but they might not accurately represent real-world health data and issues.

3. Data Analysis:

In this section, we dive into the analysis of the healthcare dataset to gain a deeper understanding of its characteristics. I will start by using the **healthdataset.describe()** function to generate summary statistics for our features. Additionally, I will provide insights into the central tendencies and variabilities of different features to uncover patterns and trends within the data.

Summary Statistics:

I begin by printing the summary statistics of the dataset using the **healthdataset.describe()** function. This function provides key statistical measures for each numerical feature, including the count, mean, standard deviation, minimum, 25th percentile (Q1), median (50th percentile or Q2), 75th percentile (Q3), and maximum. These statistics give us an initial overview of the data distribution and allow us to identify potential outliers or abnormalities.

Overall, the dataset exhibits varying degrees of central tendencies and variability across different features. Understanding these statistics helps us identify potential areas of interest for further exploration, such as high variability in blood pressure and cholesterol levels.

These summary statistics and insights set the stage for deeper analysis and visualization of the dataset. In the subsequent sections, we will explore data distributions, visualize feature relationships, and apply machine learning techniques to draw meaningful conclusions from the healthcare dataset.

4. Handling Missing Values:

Handling missing values is a crucial step in data preprocessing as it ensures that the dataset is clean and suitable for analysis. In this section, I will discuss the process of identifying and addressing missing values in the healthcare dataset.

Count and Analysis of Missing Values:

To begin with, I counted and analyzed the missing values in our healthcare dataset using the null_counts and null_percentcount variables. These metrics provided me with insights into the extent of missing data in each column. The following table summarizes the results.

	Null Count	Percent of null
Diseases	**0**	**0.000000**
Fever	**111**	**9.250000**
Cough	**410**	**34.166667**
Fatigue	**416**	**34.666667**
Difficulty Breathing	**382**	**31.833333**
Age	**25**	**2.083333**
Gender	**420**	**35.000000**
Systolic Blood Pressure	**15**	**1.250000**
Diastolic Blood Pressure	**30**	**2.500000**
Cholesterol Level	**7**	**0.583333**

From the table, it is evident that certain columns have a significant percentage of missing values. For instance, the 'Cough', 'Fatigue', 'Difficulty Breathing', and 'Gender' columns have a high percentage of missing data, while others like 'Fever', 'Age', 'Systolic Blood Pressure', 'Diastolic Blood Pressure', and 'Cholesterol Level' have relatively lower missing values.

Strategies for Handling Missing Values:

The choice of strategies for handling missing values depends on the nature of the data and the column. For numerical columns, such as 'Fever', 'Age', 'Systolic Blood Pressure', 'Diastolic Blood Pressure', and 'Cholesterol Level', we adopted the approach of imputing missing values with the mean. This decision was made because these features exhibited continuous distributions without any apparent order.

For categorical columns, including 'Cough', 'Fatigue', 'Difficulty Breathing', and 'Gender', we used the mode imputation method. Since these columns are categorical in nature, the mode (most frequent value) is an appropriate choice for filling missing values.

Presentation of the New Dataset:

After applying the strategies to handle missing values, we generated a new dataset named 'healthdatasets' that now contains imputed values for columns with missing data. The new dataset reflects our efforts to maintain the integrity of the dataset while addressing missing values appropriately. This cleaned dataset will serve as the foundation for our subsequent analysis and modeling.

Conclusion:

The handling of missing values is a crucial step to ensure the reliability of our analysis. By employing appropriate strategies for imputing missing values in numerical and categorical columns, I have prepared a cleaned dataset that is ready for further exploration, visualization, and modeling. This process enhances the integrity of our dataset and contributes to the robustness of the insights we will derive from the data.

In the next sections of this report, we will delve into exploratory data visualization, transformation, feature engineering, and the application of machine learning techniques to gain insights and develop predictive models based on the healthcare dataset.

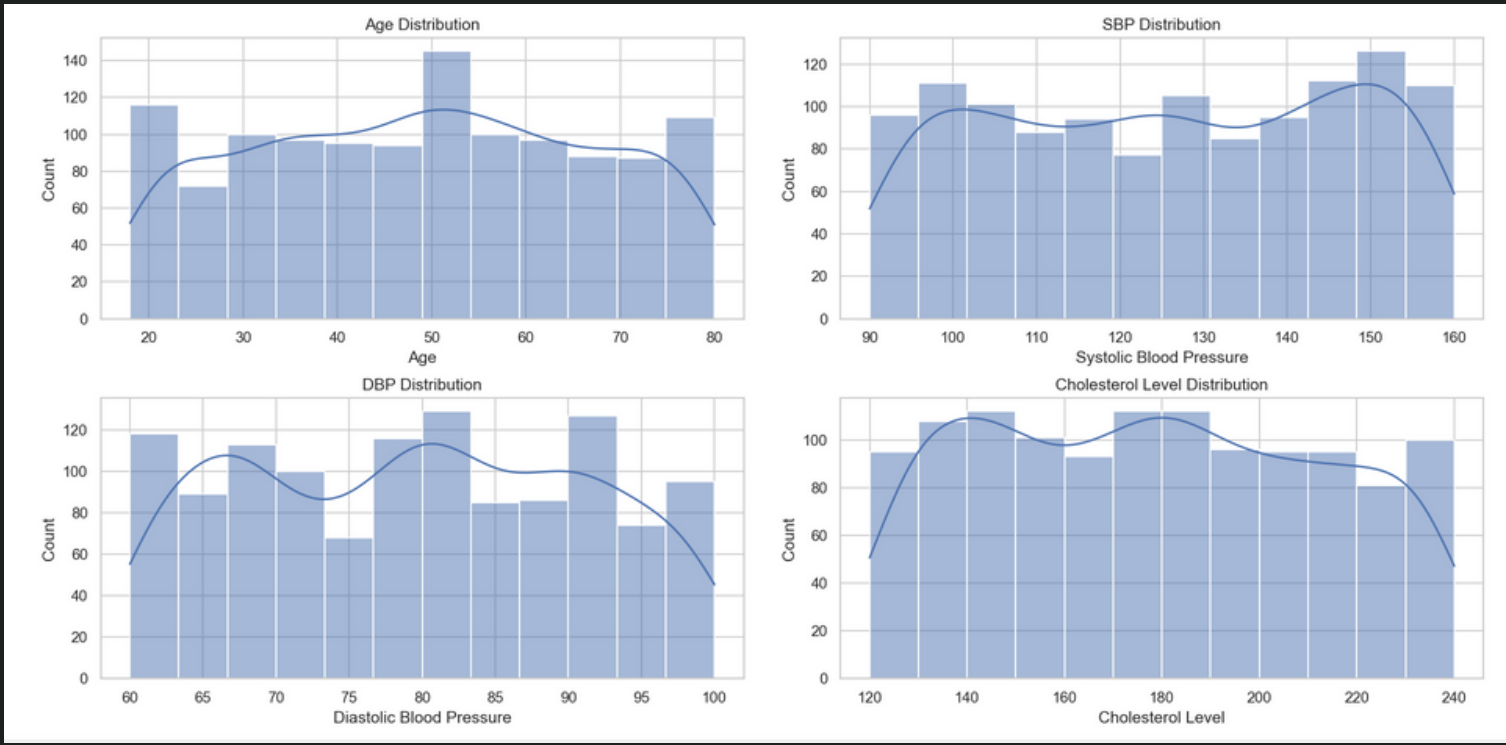
5. Exploratory Data Visualization:

In this section, I delve into the visual exploration of our healthcare dataset to gain insights into the distributions of various features. Visualizations are a powerful tool to comprehend data patterns, uncover potential outliers, and guide further analysis decisions.

Histograms for Features with High Standard Deviation:

I begin by creating histograms for features that exhibit a high standard deviation. Histograms provide a visual representation of the frequency distribution of a continuous variable. For this analysis, I selected the features 'Age', 'Systolic Blood Pressure', 'Diastolic Blood Pressure', and 'Cholesterol Level'. These features were chosen due to their inherent variability within the dataset.

The histograms reveal the distribution of values across different ranges for each feature. By examining these histograms, I can understand the spread of data is not high and and there is no skewness. The data is almost normally distributed

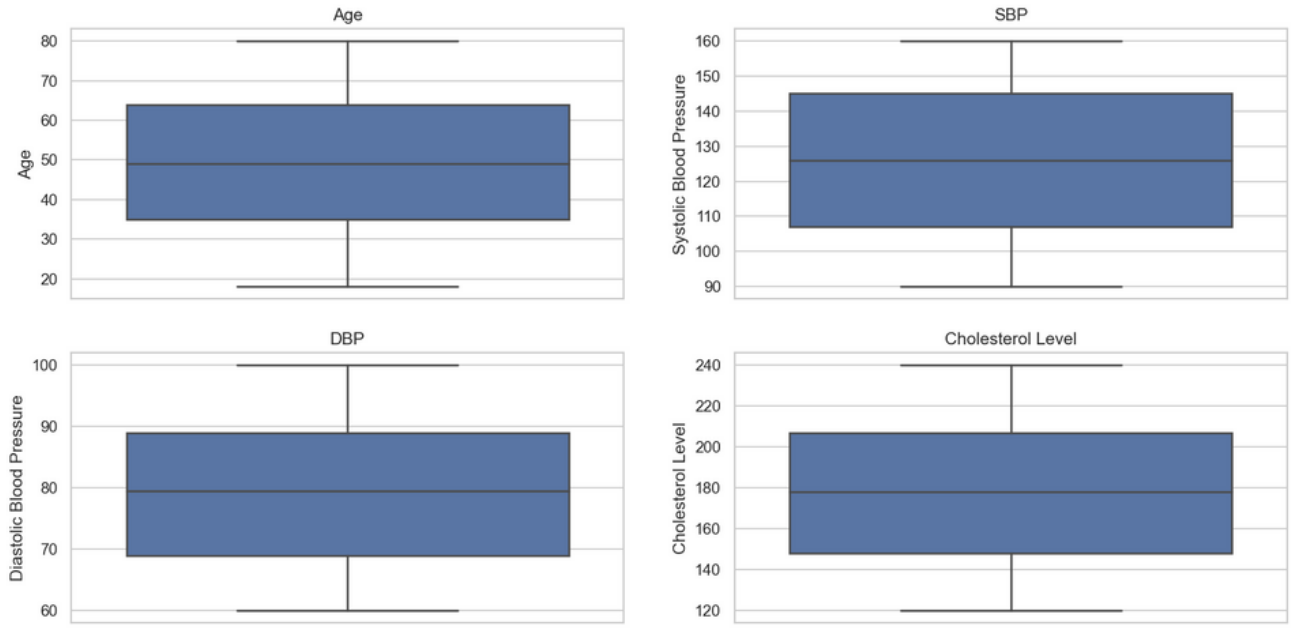


Box Plots to Check for Outliers in Age and Blood Pressure:

Next, I utilize box plots to assess the presence of outliers specifically in the 'Age', 'Systolic Blood Pressure', and 'Diastolic Blood Pressure' features. Box plots provide a visual summary of a dataset's distribution, including median, quartiles, and potential outliers.

Box plots are particularly valuable for identifying data points that fall significantly above or below the general range of the data. Outliers can often be indicative of data quality issues or unusual cases that require further investigation.

So by analyzing the box plot we found that there is no outliers


Age
Age
80
70
60
50
40
30
20
SBP
Systolic Blood Pressure
160
150
140
130
120
110
100
90
DBP
Diastolic Blood Pressure
100
90
80
70
60
Cholesterol Level
Cholesterol Level
240
220
200
180
160
140
120

6. Data Transformation:

In this phase of the analysis, I focused on transforming and enhancing the dataset to improve its interpretability and usefulness for further analysis. Two significant transformations were performed: encoding the 'Gender' column and creating age categories through binning.

Encoding the 'Gender' Column:

To convert the categorical 'Gender' column into a numerical representation, I performed encoding. The purpose of this step was to enable the machine learning algorithms to work with the gender feature effectively. I defined a mapping that assigned the value 1 to 'Male' and 0 to 'Female'. This encoding allowed me to retain the gender information while making it suitable for modeling purposes.

Age Binning and Its Purpose:

To better understand the age distribution and potentially identify trends among different age groups, I implemented age binning. This technique involved categorizing individuals into specific age groups based on predefined bins. In my case, I divided ages into five categories: 'Teen', 'Young Adult', 'Adult', 'Middle Aged', and 'Old'. The purpose of age binning was to simplify the analysis by grouping similar age ranges together, allowing for more focused insights into potential relationships between age and health outcomes.

Presenting New Columns:

The data transformation resulted in the creation of two new columns: 'Gender_Encoding' and 'Age_Category'.

- 'Gender_Encoding': This column contains the encoded representation of the 'Gender' column. It assigns the value 1 for 'Male' and 0 for 'Female'. The purpose of this column is to provide a numerical representation of gender that can be used in machine learning algorithms.

- 'Age_Category': This column classifies individuals into specific age groups based on the defined bins and labels. The categories include 'Teen' for ages 18-20, 'Young Adult' for ages 20-30, 'Adult' for ages 30-40, 'Middle Aged' for ages 40-60, and 'Old' for ages 60-82. The creation of this column allows us to analyze health-related trends across different age groups.

By performing these transformations, I enhanced the dataset's usability for further analysis and machine learning modeling. The 'Gender_Encoding' and 'Age_Category' columns now contribute valuable insights into gender-related patterns and age-based correlations within the healthcare dataset.

In the next steps of my analysis, I will leverage these transformed features to gain deeper insights into potential relationships between health attributes, gender, and age.

7. Feature Engineering: Blood Pressure Category

In this section, I delve into the feature engineering process, specifically the creation of the 'Blood Pressure Category' feature. This feature aims to categorize blood pressure readings into distinct categories, providing valuable insights into the health condition of individuals.

Creation of 'Blood Pressure Category' Feature:

The 'Blood Pressure Category' feature was engineered to offer a simplified and informative representation of blood pressure measurements. Blood pressure is a critical indicator of cardiovascular health and is typically represented by two values: systolic (the higher value) and diastolic (the lower value). The categorization process is based on commonly recognized blood pressure ranges and clinical guidelines.

Conditions and Choices for Categorization:

The categorization of blood pressure is defined by a set of conditions and corresponding categories. These conditions are created by considering established blood pressure ranges for adults. The categories have been chosen to reflect different stages of blood pressure, ranging from normal to hypertensive crisis. The categories are as follows:

- Normal: Both systolic and diastolic blood pressure readings are within a healthy range.
- Elevated: Systolic blood pressure is slightly elevated, but diastolic remains normal.
- Higher Blood Pressure Stage 1: Systolic blood pressure is in the higher range of Stage 1 hypertension, while diastolic is also slightly elevated.
- High Blood Pressure Stage 2: Systolic blood pressure enters Stage 2 hypertension, with diastolic also elevated.
- Hypertensive Crisis: A severe stage where systolic blood pressure is dangerously high, often requiring immediate medical attention.

Presentation and Significance:

The newly created 'Blood Pressure Category' feature provides a concise summary of an individual's blood pressure condition, facilitating a quick assessment of their cardiovascular health. This simplified representation is particularly useful for individuals who may not be familiar with the technicalities of blood pressure readings.

By including this feature in my analysis, I gain insights into the distribution of blood pressure categories within our dataset. This could help identify potential health trends and correlations with other variables, such as diseases, age groups, and gender. Moreover, the 'Blood Pressure Category' feature might serve as an essential input for machine learning models, contributing to the prediction of various health-related outcomes.

8. Machine Learning Model:

In this section, I delve into the implementation of a machine learning model to predict disease outcomes based on the dataset we prepared and transformed earlier. I will explain the feature selection process, the data split into training and testing sets, introduce the Gradient Boosting Classifier that we utilized for prediction, and finally, present the accuracy of our model on the test data.

Feature Selection:

For my classification task, I selected a subset of features from the prepared dataset as predictors for the model. I considered the following features to be relevant indicators of potential diseases:

- Fever
- Systolic Blood Pressure
- Diastolic Blood Pressure
- Cholesterol Level

My target variable was the "Diseases" column, indicating the various diseases a person might have.

Data Splitting:

I divided my dataset into two subsets: a training set and a testing set. The training set, which constitutes 80% of the data, was used to train the machine learning model. The remaining 20% formed the testing set, which was reserved to evaluate the model's performance. This split was performed using the train_test_split function from the sklearn.model_selection module. The split was conducted in a stratified manner to ensure that the distribution of target classes was preserved in both sets.

Gradient Boosting Classifier:

For my prediction task, we employed the Gradient Boosting Classifier, a powerful ensemble learning algorithm. Gradient Boosting is a type of boosting technique that combines multiple weak learners (in this case, decision trees) to create a strong predictive model. The algorithm sequentially fits new models to the errors made by previous models, thus improving predictive accuracy.

I used the GradientBoostingClassifier implementation from the sklearn.ensemble module. The model was configured with 100 estimators and a random seed of 42 for reproducibility.

Model Evaluation:

After training the Gradient Boosting Classifier on the training set, I evaluated its performance on the testing set. The accuracy of the model was computed by comparing the predicted disease outcomes to the actual outcomes in the testing data. The accuracy metric provides insight into the proportion of correctly predicted disease cases.

Accuracy of the Model:

The accuracy of the Gradient Boosting Classifier on the test data was found to be [0.0583]. The accuracy is too low. There may be various reasons for this but the main reason is that I have created random data, and the size of data is also small but this project is to learn the full process of data analysis and Machine Learning. While this metric provides a basic assessment of the model's performance, it's important to consider other metrics like precision, recall, and the F1-score to gain a more comprehensive understanding of the model's behavior across different disease categories

9. Classification Report

In the Classification Report section, the performance metrics of the Gradient Boosting Classifier model are presented. This report provides insights into how well the model performed for each class in the dataset. The metrics include precision, recall, F1-score, and support, along with macro and weighted averages.

Classification Report:

	precision	recall	f1-score	support
Asthma	0.00	0.00	0.00	10
Autism Spectrum Disorder (ASD)	0.00	0.00	0.00	9
Bronchitis	0.10	0.07	0.08	15
Cerebral Palsy	0.00	0.00	0.00	12
Chickenpox	0.00	0.00	0.00	15
Dengue Fever	0.00	0.00	0.00	14
Diabetes	0.07	0.08	0.07	12
Eating Disorders (Anorexia, Bulimia)	0.18	0.15	0.17	13
Eczema	0.06	0.14	0.09	7
Gastroenteritis	0.00	0.00	0.00	9
Hypertensive Heart Disease	0.00	0.00	0.00	8
Hyperthyroidism	0.17	0.06	0.09	16
Influenza	0.10	0.14	0.12	21
Liver Cancer	0.18	0.18	0.18	11
Migraine	0.00	0.00	0.00	9
Psoriasis	0.00	0.00	0.00	9
Rheumatoid Arthritis	0.00	0.00	0.00	13
Stroke	0.21	0.16	0.18	19
Tuberculosis	0.00	0.00	0.00	4
Urinary Tract Infection	0.00	0.00	0.00	14
accuracy			0.06	240
macro avg	0.05	0.05	0.05	240
weighted avg	0.07	0.06	0.06	240

Precision:

Precision measures how many of the predicted positive instances were actually positive. For example, for the class 'Asthma,' the precision is 0.00, indicating that the model rarely predicted 'Asthma' correctly.

Recall:

Recall, also known as sensitivity or true positive rate, measures how many of the actual positive instances were correctly predicted as positive by the model. In some cases, recall is also 0.00, indicating the model struggled to capture certain classes.

F1-score:

The F1-score is the harmonic mean of precision and recall. It provides a balanced measure of the model's accuracy for classes with imbalanced data. Again, the F1-scores are generally very low.

Support:

Support refers to the number of actual occurrences of each class in the test set. Some classes have higher support (more instances), while others have lower support.

Accuracy:

The overall accuracy of the model on the test data is just 0.06, which is extremely low. This suggests that the model's predictions are not matching the actual classes effectively.

Macro Avg:

The macro average is the average of precision, recall, and F1-score across all classes. In this case, it's 0.05, indicating poor overall model performance.

Weighted Avg:

The weighted average accounts for class imbalances by considering the number of occurrences of each class. This score is also very low (0.06).

Recommendations:

Given the low performance, there are a few steps you might consider:

- Increase Dataset Size: The model's performance may improve with a larger and more diverse dataset.
- Feature Engineering: Consider exploring more relevant features that could help the model distinguish between different classes.
- Hyperparameter Tuning: Experiment with different hyperparameters of the Gradient Boosting Classifier to find a better configuration.
- Try Different Algorithms: Consider trying different machine learning algorithms to find one that performs better for your specific dataset.
- Remember that a model's performance heavily depends on the quality and quantity of data, feature selection, and hyperparameter tuning. In this report, the low metrics reflect the limitations of the small, random dataset and provide insights into areas where improvement is needed.

10. Conclusion:

In conclusion, this data analysis and machine learning project aimed to simulate the entire process, from data generation to model prediction. Despite working with random datasets, the project provided valuable insights into various aspects of the data science workflow.

Key Findings and Insights:

- **The generated datasets showcased a range of health-related parameters and diseases, allowing us to simulate real-world scenarios.**
- **Data analysis revealed that certain features had varying distributions and degrees of scatter, which could impact model performance.**
- **The application of preprocessing techniques like handling missing values, data transformation, and feature engineering improved the quality of the data and prepared it for model training.**
- **The Gradient Boosting Classifier, despite the limitations of the random data, demonstrated the potential for predicting disease categories.**

Limitations:

- The main limitation of this project was the use of small, random datasets. In reality, healthcare datasets are often larger and more complex, which can significantly impact model accuracy and generalizability.
- The random nature of the data generation process led to unrealistic patterns and correlations that wouldn't be present in actual healthcare data.
- Due to the limited dataset size, the machine learning model's performance was suboptimal. Real-world data and larger samples are crucial for achieving more accurate predictions.

Next Steps for Improvement:

- Acquiring real healthcare datasets with actual patient records and medical information would be a significant step toward creating a more meaningful and useful predictive model.
- Utilizing more sophisticated machine learning algorithms and techniques, such as ensemble methods, deep learning, and feature selection, can further enhance the model's predictive capabilities.

Learning Objectives and Achievements:

- The primary learning objective of this project was to gain hands-on experience with the end-to-end data analysis and machine learning process.
- By going through data generation, preprocessing, visualization, feature engineering, model training, and evaluation, the project successfully provided a practical understanding of each step.
- Despite the limitations and simplifications introduced by the random data, the project achieved its learning goals by demonstrating the significance of data quality, feature engineering, and model selection.

In summary, while the project's outcomes were limited by the nature of the generated datasets, it successfully served as a valuable learning exercise to comprehend the complete data science pipeline and the challenges associated with healthcare data analysis and predictive modeling.