**Healthcare Test Results Classification using Machine Learning**

Presented To: Dr Manal Tantawy

|  |  |
| --- | --- |
| **Name** | **ID** |
| Yahia Tamer | 2022/02264 |
| Nouran Hassan | 2022/00062 |
| Malak Mohamed | 2022/07005 |
| Roaa Khaled | 2022/05885 |

Contents

[**1. Introduction** 3](#_Toc199112047)

[**2. Dataset Description** 3](#_Toc199112048)

[3**. Exploratory Data Analysis (EDA)** 4](#_Toc199112049)

[3.1 Summary Statistics 4](#_Toc199112050)

[3.2 Check null values in the dataset 4](#_Toc199112051)

[3.3 Categorical Features Analysis 4](#_Toc199112052)

[3.4 Check Duplicates 4](#_Toc199112053)

[3.5 Check Target Variable Distribution 4](#_Toc199112054)

[3.6 Numerical Features Analysis 5](#_Toc199112055)

[3.7 Correlation Matrix 7](#_Toc199112056)

[3.8 Outliers Detection 7](#_Toc199112057)

[**4. Data Preprocessing** 7](#_Toc199112058)

[4.1 Handling Missing Values 7](#_Toc199112059)

[4.2 Encoding Categorical Variables 7](#_Toc199112060)

[4.3 Feature Scaling 8](#_Toc199112061)

[4.4 Drop Columns 8](#_Toc199112062)

[4.5. Principal Component Analysis 8](#_Toc199112063)

[**5. Machine Learning Models** 9](#_Toc199112064)

[5.1 Logistic Regression 9](#_Toc199112065)

[5.2 MLP 10](#_Toc199112066)

[5.3 Naïve Bayes 10](#_Toc199112067)

[5.3.2 Naïve Bayes (No PCA) 11](#_Toc199112068)

[5.4 Random Forest 11](#_Toc199112069)

[5.5 Support Vector Machine 11](#_Toc199112070)

[5.6 XGBoost 12](#_Toc199112071)

# **1. Introduction**

This project focuses on classifying healthcare test results into three categories: **Normal**, **Abnormal**, and **Inconclusive** using machine learning techniques. A complete pipeline was developed, including data preprocessing, dimensionality reduction with PCA, and training various classification models. The goal is to evaluate and compare the performance of these models with and without PCA, using metrics such as accuracy, precision, recall, F1-score, and confusion matrices. The results help identify the most effective approach for accurate and reliable classification in a healthcare context.

# **2. Dataset Description**

|  |  |
| --- | --- |
| Feature | Description |
| Name | Name of the patient associated with the healthcare record. |
| Age | Age of the patient at the time of admission (in years). |
| Gender | Gender of the patient ("Male" or "Female"). |
| Blood Type | Patient's blood type (e.g., "A+", "O-", etc.). |
| Medical Condition | Primary medical condition or diagnosis (e.g., "Diabetes", "Hypertension", "Asthma"). |
| Date of Admission | Date the patient was admitted to the healthcare facility. |
| Doctor | Name of the doctor responsible for the patient's care. |
| Hospital | Name of the healthcare facility or hospital where the patient was admitted. |
| Insurance Provider | Patient's insurance provider (e.g., "Aetna", "Blue Cross", "Cigna", "UnitedHealthcare", "Medicare"). |
| Billing Amount | Amount billed for healthcare services (floating-point number). |
| Room Number | Room number where the patient stayed during admission. |
| Admission Type | Type of admission: "Emergency", "Elective", or "Urgent". |
| Discharge Date | Date the patient was discharged from the healthcare facility. |
| Medication | Medication prescribed/administered during admission (e.g., "Aspirin", "Penicillin", "Lipitor"). |
| Test Results | Result of medical tests: "Normal", "Abnormal", or "Inconclusive". |

# **3. Exploratory Data Analysis (EDA)**

### 3.1 Summary Statistics

### 

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Statistic** | **ID** | **Age** | **Billing Amount** | **Room Number** |
| count | 50000.000000 | 50000.000000 | 49276.000000 | 50000.000000 |
| mean | 25000.500000 | 45.863700 | 5397.314987 | 282.999980 |
| std | 14433.901067 | 24.416119 | 3240.601901 | 109.993534 |
| min | 1.000000 | 0.000000 | -995.211704 | 50.000000 |
| 25% | 12500.750000 | 26.000000 | 2645.811489 | 203.000000 |
| 50% | 25000.500000 | 45.000000 | 5313.507889 | 286.000000 |
| 75% | 37500.250000 | 65.000000 | 8027.205893 | 370.000000 |
| max | 50000.000000 | 100.000000 | 12635.764460 | 500.000000 |

### 3.2 Check null values in the dataset

|  |  |
| --- | --- |
| **Column** | **Missing Values** |
| Blood Type | 3,065 |
| Doctor | 804 |
| Hospital | 959 |
| Insurance Provider | 293 |
| Billing Amount | 724 |
| Admission Type | 181 |

Table 1: Features with null values

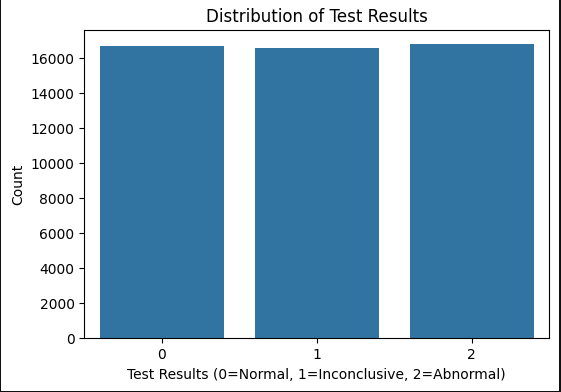
### 3.3 Categorical Features Analysis

### 3.4 Check Duplicates

### 3.5 Check Target Variable Distribution

Analysis:

* Class Balance of Test Results (Normal/Inconclusive/Abnormal)
* Potential Bias Detection

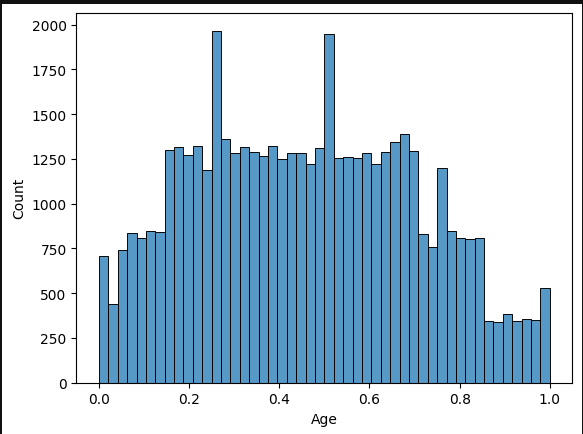


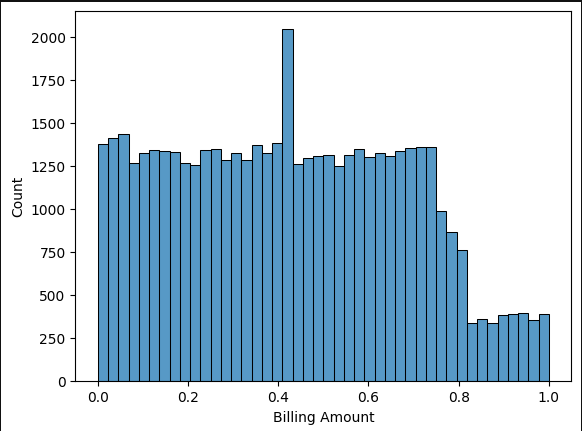
1: Target Class Imbalance

### 3.6 Numerical Features Analysis

Features: Age, Billing Amount, Days Spent

| **Feature** | **Mean** | **Std Dev** | **Skewness** | **Outliers (IQR)** |
| --- | --- | --- | --- | --- |
| Age | 45.86 | 24.42 | 0.12 | 0 |
| Billing Amount | 5,397.31 | 3,217.05 | 0.05 | 0 |



2: Histogram Plot for Age Feature

3: Histogram Plotting of Billing Amount Feature

### **3.7 Correlation Matrix**

A correlation matrix is a table showing how strongly pairs of variables are related, with values from -1 (negative) to 1 (positive). It helps identify patterns or relationships between variables.

|  |  |
| --- | --- |
| **Feature** | **Correlation** |
| Test Results | 1.000000 |
| Admission Type\_Urgent | 0.489156 |
| Insurance Provider\_Blue Cross | 0.482841 |
| Room Number | 0.283464 |
| Medical Condition\_Asthma | 0.186281 |
| Blood Type\_AB- | 0.130850 |
| Medical Condition\_Diabetes | 0.128517 |
| Blood Type\_O+ | 0.093200 |
| Gender\_Male | 0.084318 |
| Blood Type\_A- | 0.082569 |
| Blood Type\_O- | 0.068833 |

Table 2: Sample of Correlation Matrix

### **3.8 Outliers Detection**

* No Outliers

# **4. Data Preprocessing**

### **4.1 Handling Missing Values**

|  |  |
| --- | --- |
| **Feature** | **Imputation Method** |
| Billing Amount | Mean |
| Blood Type | Mode |
| Admission Type | Mode |
| Doctor | Mode |
| Insurance Provider | Mode |
| Hospital | Mode |

### **4.2 Encoding Categorical Variables**

|  |  |  |  |
| --- | --- | --- | --- |
| **Feature** | **Values** | **# Columns After Encoding** | **Encoding Type** |
| Gender | 2 (Male, Female) | 2 | One-Hot Encoding |
| Blood Type | 8 (A+, A-, ..., O+) | 8 | One-Hot Encoding |
| Admission Type | 3 (Urgent, Emergency, Elective) | 3 | Label Encoding |
| Medical Condition | 6 (Asthma, Cancer, ...) | 6 | One-Hot Encoding |
| Insurance Provider | 5 (Cigna, Blue Cross, ...) | 5 | One-Hot Encoding |
| Medication | 5 (Ibuprofen, Penicillin, ...) | 5 | One-Hot Encoding |
| Test Results (Target) | 3 (Normal, Inconclusive, Abnormal) | 1 | Label Encoding |

### **4.3 Feature Scaling**

|  |  |
| --- | --- |
| **Feature** | **Scaling Method** |
| Age | Min-Max Scaler |
| Billing Amount | Min-Max Scaler |
| Days Spent | Min-Max Scaler |

### **4.4 Drop Columns**

To simplify the dataset and remove high-cardinality or irrelevant features, several columns were dropped. Additionally, a new feature Days Spent was derived from date calculations.

|  |  |
| --- | --- |
| **Column Dropped** | **Reason** |
| ID | Unique identifier, not useful for prediction |
| Name | High-cardinality, non-numeric, and irrelevant to classification |
| Room Number | High-cardinality with no significant correlation to target |
| Date of Admission | Replaced with Days Spent |
| Discharge Date | Replaced with Days Spent |
| Doctor |  |
| Hospital |  |

### 4.5. Principal Component Analysis

Principal Component Analysis (PCA) was used to reduce the dataset’s dimensionality while preserving most of its variance.

*Steps:*

**Data Preparation:**  
Boolean columns were converted to integers for compatibility.

**PCA Application:**  
PCA was applied with 25 components, capturing key variance.

**Explained Variance:**  
PC1 to PC25 explained from 11.15% to 1.27% of the variance each, collectively covering a significant portion.

**Target Merge & Save:**  
The 'Test Results' column was reattached, and the final dataset was saved as PCA\_Training\_Set\_Preprocessed\_Final.csv.

PCA improved model efficiency by reducing redundant features while retaining essential data patterns.

# **5. Machine Learning Models**

### 5.1 Logistic Regression

Logistic Regression is a linear classification algorithm widely used for binary and multiclass classification tasks. It models the probability that a given input belongs to a particular class using the logistic (sigmoid) function. Despite its simplicity, logistic regression can perform competitively when features are informative and linearly separable

#### 5.1.1 Logistic Regression (PCA)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| 0 | 0.79 | 0.78 | 0.78 | 2547 |
| 1 | 0.73 | 0.72 | 0.72 | 2440 |
| 2 | 0.80 | 0.82 | 0.81 | 2513 |
| Accuracy |  |  | 0.77 | 7500 |
| Macro Avg | 0.77 | 0.77 | 0.77 | 7500 |
| Weighted Avg | 0.77 | 0.77 | 0.77 | 7500 |

Table 3: LR Test Classification Report (PCA)

#### 5.1.2 Logistic Regression (No PCA)

|  |  |
| --- | --- |
| Metric | Value |
| Precision | 0.773936 |
| Sensitivity | 0.774400 |
| F1 Score | 0.774107 |
| Accuracy | 0.774400 |

Table 4:Performance Metrics

### 5.2 MLP

The Multilayer Perceptron (MLP) is a type of neural network with fully connected layers that can learn complex patterns through backpropagation. It is well-suited for classification tasks and performs well on non-linear data. In this project, MLP was applied both with and without PCA for dimensionality reduction.

#### 5.2.1 MLP (No PCA)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| Class 0 | 0.84 | 0.85 | 0.85 | 2547 |
| Class 1 | 0.76 | 0.75 | 0.75 | 2440 |
| Class 2 | 0.79 | 0.79 | 0.79 | 2513 |
| Macro Avg | 0.80 | 0.80 | 0.80 | 7500 |
| Weighted Avg | 0.80 | 0.80 | 0.80 | 7500 |
| Accuracy | — | — | 0.7989 | 7500 |

Table 5: MLP Test Classification Report (No PCA)

#### 5.2.2 MLP (PCA)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| Class 0 | 0.87 | 0.91 | 0.89 | 2547 |
| Class 1 | 0.79 | 0.82 | 0.81 | 2440 |
| Class 2 | 0.88 | 0.81 | 0.84 | 2513 |
| Macro Avg | 0.85 | 0.85 | 0.85 | 7500 |
| Weighted Avg | 0.85 | 0.85 | 0.85 | 7500 |
| Accuracy | — | — | 0.8461 | 7500 |

Table 6: MLP Test Classification Report (PCA)

### 5.3 Naïve Bayes

Naive Bayes classifiers are probabilistic models based on Bayes’ theorem, assuming feature independence. Despite this strong assumption, they often perform well in classification tasks, especially with large datasets and text classification problems, due to their simplicity and efficiency.

#### 5.3.1 Naïve Bayes (PCA)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| 0 | 0.78 | 0.78 | 0.78 | 2547 |
| 1 | 0.69 | 0.71 | 0.70 | 2440 |
| 2 | 0.80 | 0.78 | 0.79 | 2513 |
| Accuracy |  |  | 0.76 | 7500 |
| Macro Avg | 0.76 | 0.76 | 0.76 | 7500 |
| Weighted Avg | 0.76 | 0.76 | 0.76 | 7500 |

Table 7:NB Test Classification Report (PCA)

### 5.3.2 Naïve Bayes (No PCA)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Class | Precision | Recall | F1-Score | Support |
| 0 | 0.79 | 0.78 | 0.78 | 2547 |
| 1 | 0.73 | 0.73 | 0.73 | 2440 |
| 2 | 0.81 | 0.83 | 0.82 | 2513 |
| Accuracy |  |  | 0.78 | 7500 |
| Macro Avg | 0.78 | 0.78 | 0.78 | 7500 |
| Weighted Avg | 0.78 | 0.78 | 0.78 | 7500 |

Table 8: NB Test Classification Report ( No PCA)

### 5.4 Random Forest

Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions to improve accuracy and control overfitting. It is robust to noise and works well for both classification and regression tasks.

#### 5.4.1 Random Forest (PCA)

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Precision | 0.862080 |
| Sensitivity | 0.861304 |
| F1 Score | 0.861516 |
| Accuracy | 0.861304 |

#### 5.4.2 Random Forest (No PCA)

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Precision | 0.862080 |
| Sensitivity | 0.861304 |
| F1 Score | 0.861516 |
| Accuracy | 0.861304 |

### 5.5 Support Vector Machine

SVM is a powerful classification algorithm that finds the optimal hyperplane separating classes by maximizing the margin between them. It works well for both linear and non-linear problems through the use of kernel functions, making it effective in high-dimensional spaces.

#### 5.5.1 Support Vector Machine (PCA)

|  |  |
| --- | --- |
| **Dataset** | **Accuracy** |
| Validation | 0.8233 |
| Test | 0.8227 |
| Training | 0.8402 |

#### 5.5.2 Support Vector Machine (No PCA)

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Precision | 0.8198 |
| Sensitivity (Recall) | 0.8203 |
| F1 Score | 0.8197 |
| Average Accuracy | 0.8203 |

### 5.6 XGBoost

XGBoost is an optimized gradient boosting algorithm that builds an ensemble of weak learners sequentially to minimize prediction errors. Known for its speed and performance, it is widely used in machine learning competitions and real-world applications.

#### 5.6.1 XGBoost (PCA)

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Precision | 0.8426 |
| Sensitivity | 0.8425 |
| F1 Score | 0.8424 |
| Accuracy | 0.8425 |

#### 5.6.2 XGBoost (No PCA)

|  |  |
| --- | --- |
| **Metric** | **Value** |
| Precision | 0.8599 |
| Sensitivity | 0.8596 |
| F1 Score | 0.8597 |
| Accuracy | 0.8596 |