Predictive Pulse: Harnessing Machine Learning for Blood Pressure Analysis

AN INDUSTRY ORIENTED MINI REPORT

Submitted to

JAWAHARLAL NEHRU TECNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

Submitted By

Gochikonda Raghavendra	21UK1A6732
Vasam Varsha	21UK1A6728
Adduri Sravani	21UK1A6741
Butti Mahender	22UK5A6706

Under the guidance of

Mrs. R. Swathi

Assistant Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VAAGDEVI ENGINEERING COLLEGE

Affiliated to JNTUH, HYDERABAD

BOLLIKUNTA, WARANGAL (T.S) – 506005

DEPARTMENT OF

COMPUTER SCIENCE AND ENGINEERING (DATA SCIENCE)

VAAGDEVI ENGINEERING COLLEGE(WARANGAL)

CERTIFICATE OF COMPLETION
INDUSTRY ORIENTED MINI PROJECT

This is to certify that the UG Project Phase-1 entitled “Predictive Pulse: Harnessing Machine Learning for Blood Pressure Analysis” is being submitted by GOCHIKONDA RAGHAVENDRA (21UK1A6732), VASAM VARSHA(21UK1A6728), ADDURI SRAVANI(21UK1A6741), BUTTI MAHENDER (22UK5A6706) in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in Computer Science & Engineering to Jawaharlal Nehru Technological University Hyderabad during the academic year 2024- 2025.

Project Guide	**HOD**
Mrs. R. Swathi	**Dr .K. Sharmila**
(Assistant Professor)	**(Professor)**

External

ACKNOWLEDGEMENT

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved **Dr.P.PRASAD RAO,** Principal, Vaagdevi Engineering College for making us available all the required assistance and for his support and inspiration to carry out this UG Project Phase-1 in the institute.

We extend our heartfelt thanks to **Dr . K. SHARMILA**, Head of the Department of CSE, Vaagdevi Engineering College for providing us necessary infrastructure and thereby giving us freedom to carry out the UG Project Phase-1.

We express heartfelt thanks to Smart Bridge Educational Services Private Limited, for their constant supervision as well as for providing necessary information regarding the UG Project Phase-1 and for their support in completing the UG Project Phase-1.

We express heartfelt thanks to the guide, **R. SWATHI,** Assistant professor, Department of CSE for his constant support and giving necessary guidance for completion of this UG Project Phase-1.

Finally, we express our sincere thanks and gratitude to my family members, friends for their encouragement and outpouring their knowledge and experience throughout the thesis.

GOCHIKONDA RAGHAVENDRA (21UK1A6732), VASAM VARSHA (21UK1A6728),ADDURI SRAVANI (21UK1A6741),BUTTI MAHENDER(22UK5A6706)

ABSTRACT

Hypertension, a prevalent and potentially life-threatening condition, necessitates accurate and continuous monitoring to manage and mitigate associated health risks. Traditional blood pressure measurement techniques often fall short in providing comprehensive insights due to their intermittent nature. This project investigates the potential of machine learning to revolutionize blood pressure analysis, aiming to improve prediction accuracy and enable early detection of hypertension.

In this project, we utilize a diverse dataset encompassing continuous blood pressure readings, patient demographics, and lifestyle factors. We employ a variety of machine learning algorithms, including

regression models, support vector machines, and neural networks, to predict blood pressure variations and identify hypertensive episodes.

TABLE OF CONTENTS:-

1.INTRODUCTION

1.1.OVERVIEW

Hypertension, commonly known as high blood pressure, is a major public health concern globally. It is a leading risk factor for cardiovascular diseases, stroke, and chronic kidney disease, significantly contributing to morbidity and mortality rates. Accurate and continuous monitoring of blood pressure is crucial for effective management and prevention of hypertensive complications. However, traditional methods of blood pressure measurement, typically performed at periodic intervals in clinical settings, often fail to capture the full spectrum of blood pressure variability and trends over time.

Recent advancements in wearable technology and digital health have made it possible to collect continuous blood pressure data, offering a more comprehensive understanding of an individual's blood pressure profile. Nevertheless, the sheer volume and complexity of the data present significant challenges for traditional analytical methods. This is where machine learning, with its ability to handle large datasets and uncover intricate patterns, presents a promising solution.

Machine learning techniques have shown great potential in various fields of healthcare, from disease prediction and diagnosis to personalized treatment recommendations. By leveraging these advanced analytical methods, it is possible to enhance the accuracy and depth of blood pressure analysis. This project aims to harness the power of machine learning to improve the prediction, monitoring, and management of blood pressure, ultimately contributing to better health outcomes.

In this project, we utilize a comprehensive dataset comprising continuous blood pressure readings, along with demographic and lifestyle information, to train and evaluate various machine learning models. Our approach involves both supervised and unsupervised learning techniques to predict blood pressure trends, detect anomalies, and identify underlying patterns. The goal is to develop a robust analytical framework that can provide personalized insights and early warnings, enabling proactive management of hypertension.

The introduction of machine learning into blood pressure analysis represents a significant advancement in healthcare, offering the potential for more precise and individualized patient care. This project not only aims to demonstrate the feasibility and effectiveness of such an approach but also to lay the groundwork for future research and clinical applications. Through this endeavor, we seek to contribute to the ongoing transformation of healthcare towards a more data-driven and patient-centric model.

1.2.PURPOSE

The primary purpose of harnessing machine learning for blood pressure analysis is to enhance the accuracy, reliability, and comprehensiveness of hypertension management. Specifically, this involves several key objectives:

1. **Improving Prediction Accuracy**: Traditional blood pressure measurements are often limited to periodic readings, which may not capture the true variability and dynamics of an individual's blood pressure. Machine learning models can analyze continuous blood pressure data to provide more accurate predictions of blood pressure trends and fluctuations.
2. **Early Detection of Hypertension**: Early detection of abnormal blood pressure patterns can lead to timely interventions, potentially preventing the progression to more severe stages of hypertension. Machine learning algorithms can identify subtle patterns and early warning signs that might be missed by conventional methods.
3. **Personalized Healthcare**: Every individual has unique risk factors and lifestyle influences that affect their blood pressure. Machine learning can analyze large and diverse datasets to provide personalized insights and recommendations, tailored to the specific needs and conditions of each patient.
4. **Uncovering Hidden Patterns**: Machine learning techniques, especially unsupervised learning, can uncover hidden patterns and correlations within the data that might not be apparent through traditional analysis. These insights can lead to a better understanding of the factors influencing blood pressure and help in the development of more effective treatment strategies.
5. **Enhancing Continuous Monitoring**: With the advent of wearable technology, continuous monitoring of blood pressure has become feasible. Machine learning can process and analyze this continuous stream of data in real-time, providing ongoing assessments and updates on an individual's blood pressure status.
6. **Reducing Healthcare Costs**: By enabling more precise and timely interventions, machine learning can help reduce the need for emergency care and hospitalizations associated with uncontrolled hypertension. This can lead to significant cost savings for both healthcare systems and patients.
7. **Advancing Research**: The integration of machine learning in blood pressure analysis contributes to the broader field of medical research by providing new methodologies and insights. This can spur further innovations and advancements in hypertension management and beyond.

2.LITERATURE SURVEY

2.1 EXISTING PROBLEM

1. **Data Quality and Availability**
    - **Problem**: High-quality, continuous blood pressure data is essential for training accurate machine learning models. However, such data is often scarce due to reliance on traditional, sporadic measurement methods and the limited adoption of continuous monitoring devices.
    - **Solution**: Increasing the use of wearable devices that provide continuous blood pressure monitoring can help generate richer datasets. Additionally, collaboration between healthcare providers and researchers can facilitate data sharing and aggregation.
2. **Data Privacy and Security**
    - **Problem**: The collection and analysis of personal health data raise significant privacy and security concerns. Patients may be reluctant to share their data due to fears of misuse or breaches.
    - **Solution**: Implementing robust data encryption and anonymization techniques can protect patient information. Clear communication about data use and benefits, along with strict adherence to data privacy regulations, can also help build patient trust.
3. **Model Interpretability and Trust**
    - **Problem**: Machine learning models, particularly deep learning algorithms, are often seen as "black boxes" due to their complexity, making it difficult for clinicians to understand and trust their predictions.
    - **Solution**: Developing interpretable machine learning models, such as decision trees or using techniques like SHAP (SHapley Additive exPlanations) values, can help elucidate model decisions. Providing clinicians with tools to visualize and explore model outputs can further enhance trust and usability.
4. **Integration with Clinical Workflows**
    - **Problem**: Integrating machine learning models into existing clinical workflows is challenging, requiring seamless interoperability with electronic health records (EHR) and other clinical systems.
    - **Solution**: Designing machine learning tools that can easily integrate with EHR systems and provide actionable insights in a clinician-friendly format is crucial. Collaboration with healthcare IT professionals can ensure smooth integration and minimize workflow disruption.
5. **Generalizability of Models**

- **Problem**: Machine learning models trained on specific datasets may not generalize well to different populations or clinical settings due to variations in demographics, measurement techniques, and clinical practices.
- **Solution**: Developing robust models through the use of diverse, multicenter datasets can improve generalizability. Cross-validation and external validation studies are also essential to test model performance across different settings.

6. **Handling Imbalanced Data**
    - **Problem**: Blood pressure data often contains imbalanced classes, with fewer instances of hypertensive episodes compared to normal readings. This can lead to biased models that perform poorly on detecting hypertension.
    - **Solution**: Techniques such as oversampling, undersampling, and synthetic data generation (e.g., SMOTE) can help balance the dataset. Additionally, using cost-sensitive learning approaches that assign higher penalties to misclassifications of the minority class can improve model performance.

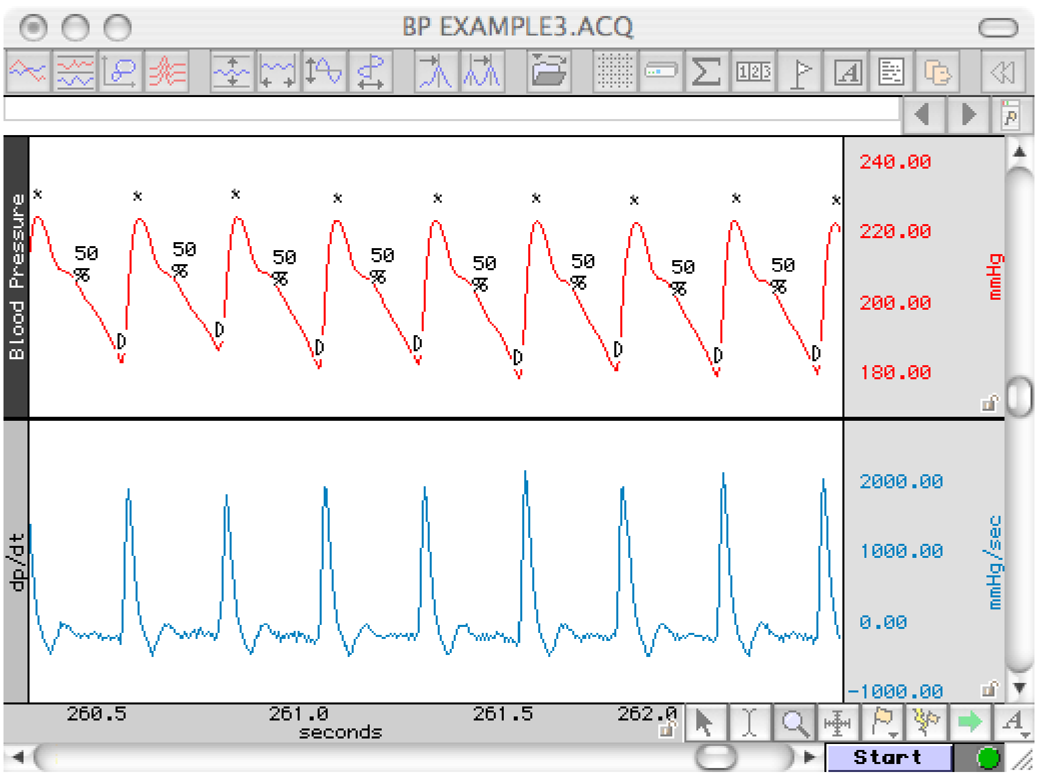
2.2 PROPOSED SOLLUTION

1. **Enhanced Data Collection Methods**
    - Wearable devices like smartwatches and fitness trackers equipped with blood pressure monitoring capabilities are becoming more prevalent. These devices can collect continuous, real-time data, providing a rich source of information for machine learning models.
2. **Advanced Preprocessing Techniques**
    - Data preprocessing techniques, such as signal filtering and feature extraction, can enhance the quality of blood pressure data. Machine learning models can benefit from engineered features that capture relevant physiological patterns and trends.
3. **Hybrid Modeling Approaches**
    - Combining different machine learning models or integrating traditional statistical methods with machine learning can improve prediction accuracy and robustness. Hybrid approaches can leverage the strengths of multiple techniques to achieve better performance.
4. **Patient-Specific Models**
    - Developing individualized models that account for patient-specific factors, such as age, gender, medical history, and lifestyle, can lead to more accurate and personalized predictions. Adaptive learning techniques that update models based on new data can further refine predictions over time.

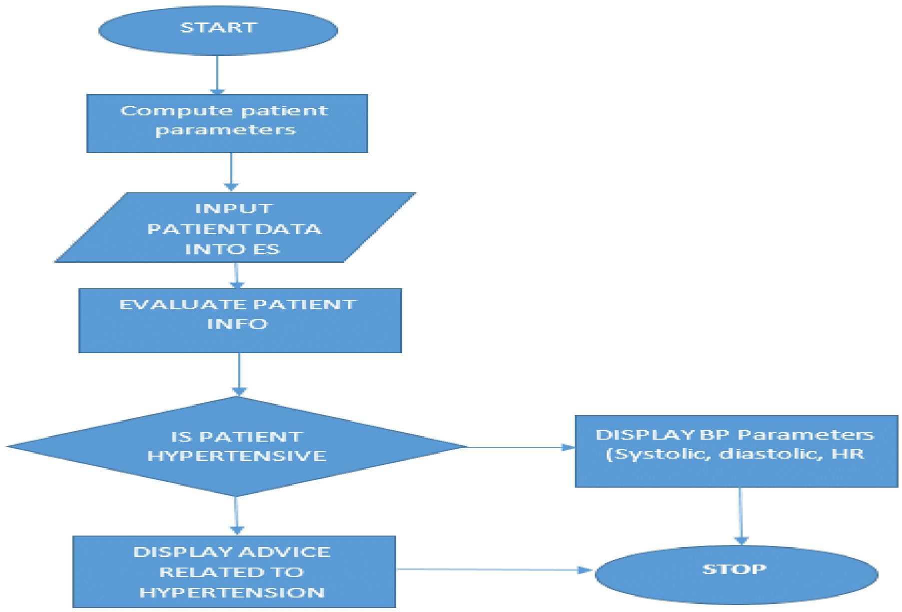
5. **Real-Time Monitoring and Feedback Systems**
   - Implementing real-time monitoring systems that provide instant feedback and alerts to patients and clinicians can facilitate proactive blood pressure management. These systems can leverage machine learning models to detect anomalies and recommend timely interventions.
6. **Collaboration and Multidisciplinary Research**
   - Collaborative efforts between data scientists, clinicians, and healthcare professionals are essential to ensure the successful application of machine learning in blood pressure analysis. Multidisciplinary research can address complex challenges and drive innovation in this field.

3.THEORITICAL ANALYSIS

3.1 BLOCK DIAGRAM

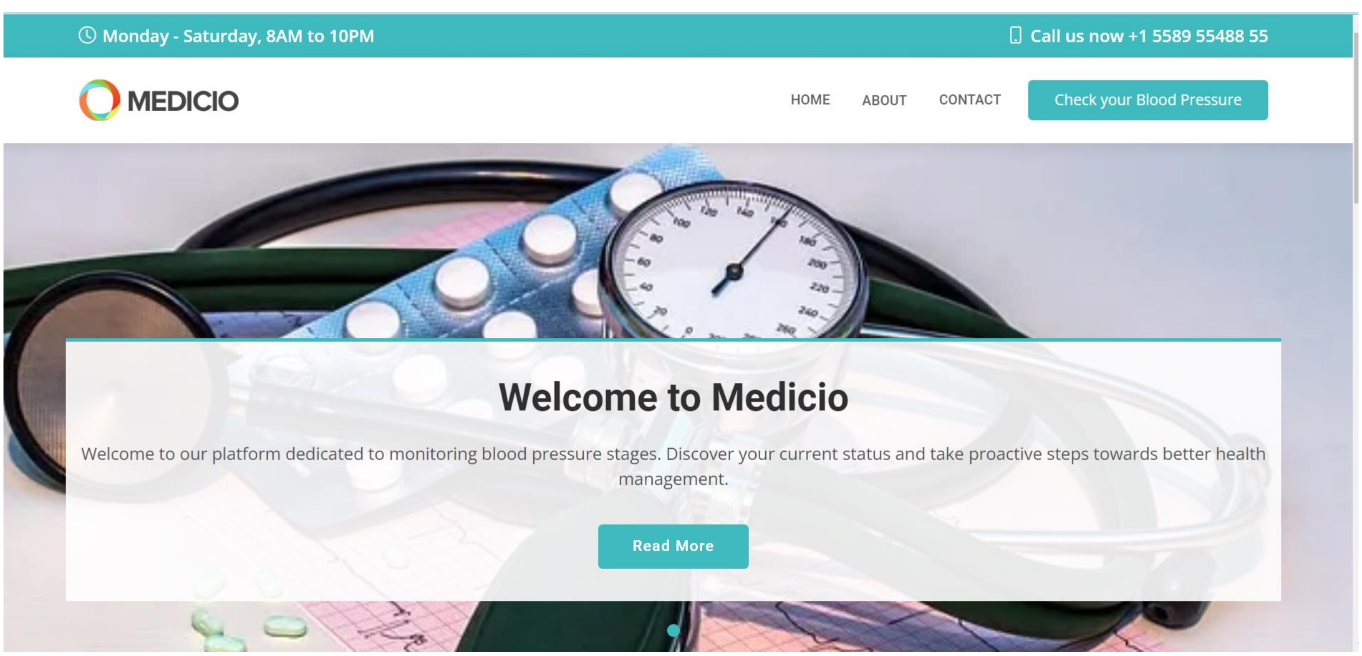


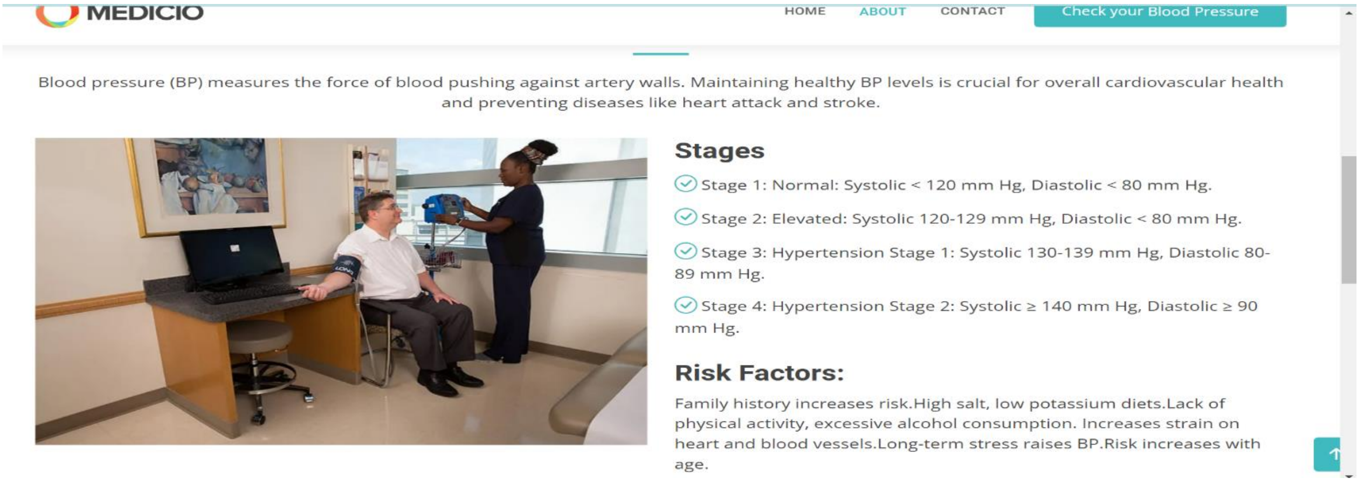
5.FLOWCHART



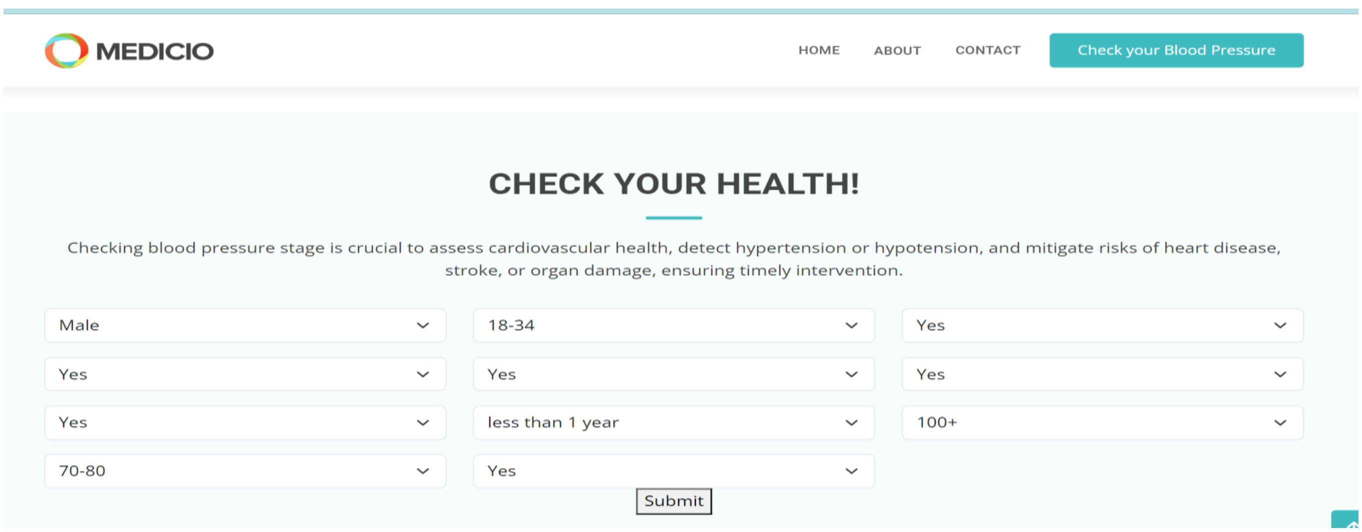
6.RESULT

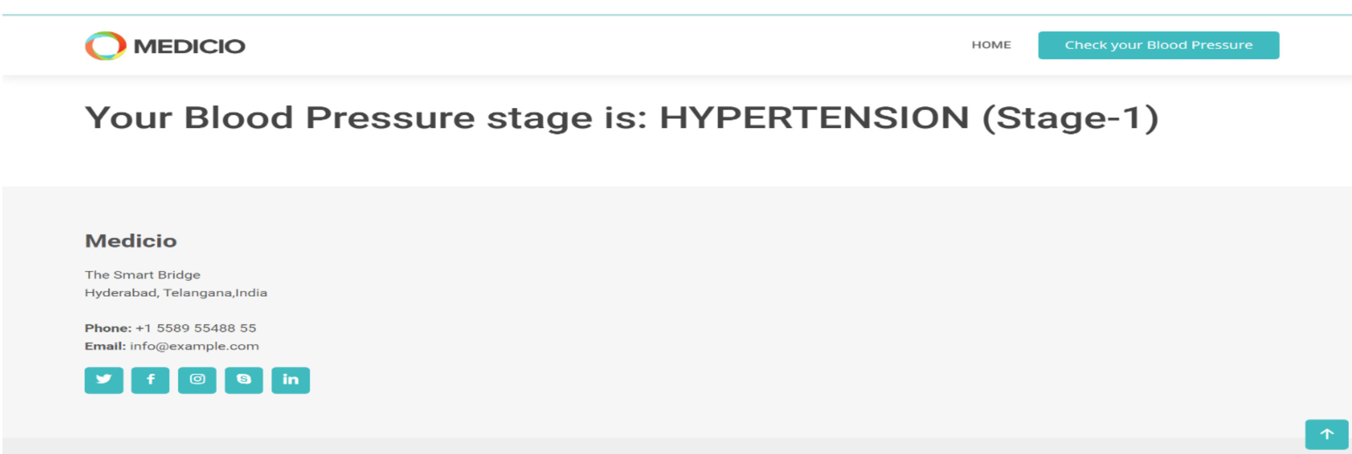
HOME PAGE



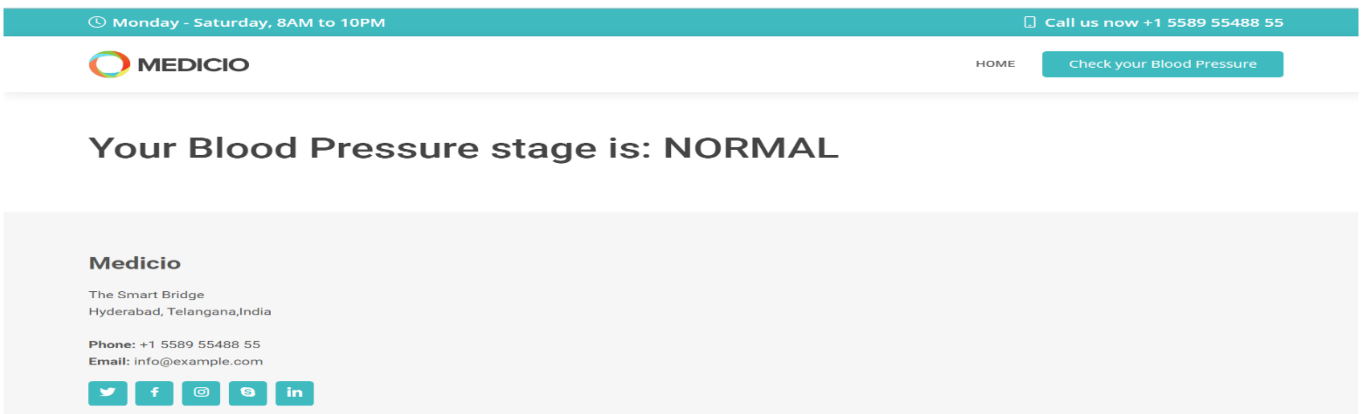


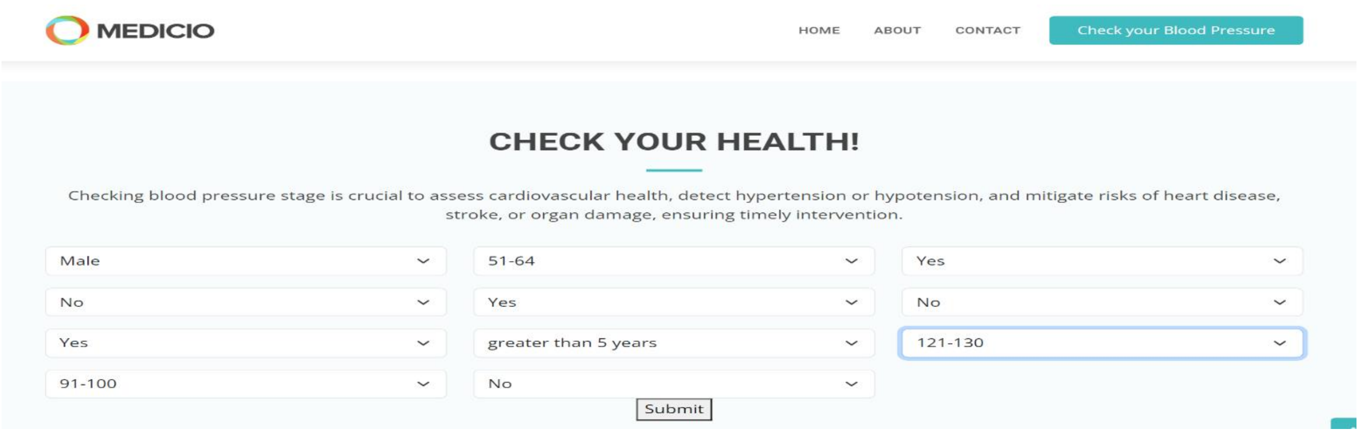
PREDICTIONS

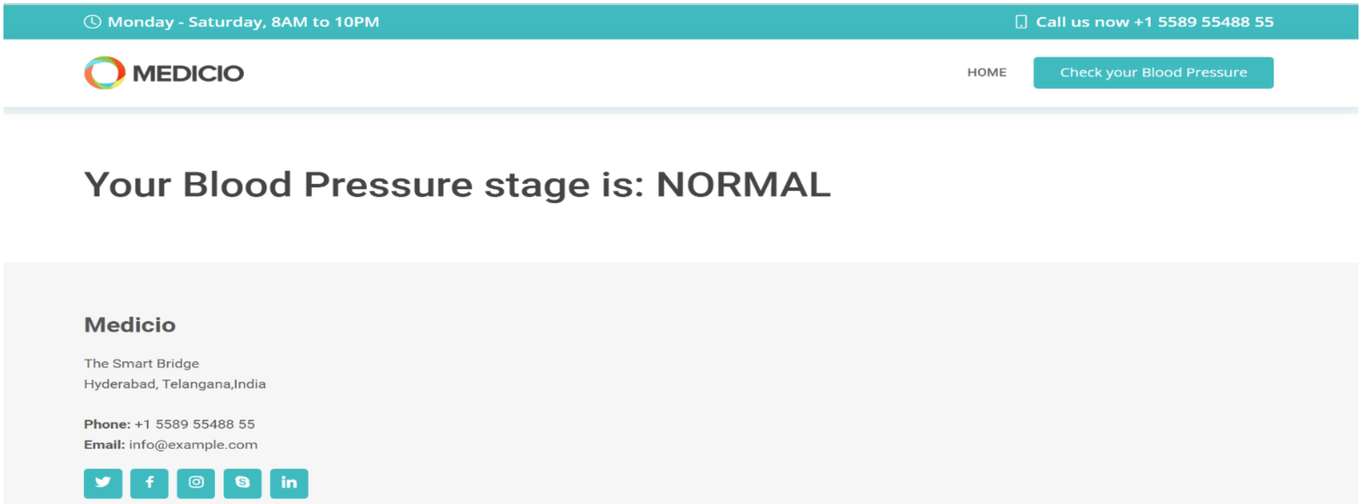





MEDICIO
HOME
ABOUT
CONTACT
Check your Blood Pressure
CHECK YOUR HEALTH!
Checking blood pressure stage is crucial to assess cardiovascular health, detect hypertension or hypotension, and mitigate risks of heart disease, stroke, or organ damage, ensuring timely intervention.
Female
35-50
No
No
No
No
No
1- 5 years
111-120
81-90
No
Submit


Monday - Saturday, 8AM to 10PM
Call us now +1 5589 55488 55
MEDICIO
HOME
Check your Blood Pressure
Your Blood Pressure stage is: NORMAL
Medicio
The Smart Bridge
Hyderabad, Telangana,India
Phone: +1 5589 55488 55
Email: info@example.com


MEDICIO
HOME
ABOUT
CONTACT
Check your Blood Pressure
CHECK YOUR HEALTH!
Checking blood pressure stage is crucial to assess cardiovascular health, detect hypertension or hypotension, and mitigate risks of heart disease, stroke, or organ damage, ensuring timely intervention.
Male
51-64
Yes
No
Yes
No
Yes
greater than 5 years
121-130
91-100
No
Submit



7.ADVANTAGES AND DISADVANTAGES

ADVANTAGES:

1. **Improved Accuracy and Precision**
   - **Benefit**: Machine learning algorithms can detect subtle patterns and correlations in large datasets that might be missed by traditional methods.
   - **Impact**: More accurate and reliable blood pressure measurements and predictions.
2. **Continuous Monitoring**
   - **Benefit**: Continuous data collection through wearable devices provides a comprehensive view of an individual's blood pressure over time.
   - **Impact**: Early detection of abnormal trends, enabling timely interventions.
3. **Personalized Insights**
   - **Benefit**: ML models can tailor insights and recommendations based on individual-specific factors like genetics, lifestyle, and medical history.
   - **Impact**: Enhanced patient engagement and adherence to personalized treatment plans.
4. **Predictive Analytics**
   - **Benefit**: Forecast future blood pressure trends and potential hypertensive episodes.
   - **Impact**: Proactive management and preventive measures, reducing the risk of complications.
5. **Anomaly Detection**
   - **Benefit**: Effective identification of deviations from normal blood pressure patterns.
   - **Impact**: Quick response to critical health events, improving patient safety.
6. **Efficient Data Handling**
   - **Benefit**: Automation of data processing and analysis.
   - **Impact**: Reduces the workload on healthcare professionals, allowing them to focus more on patient care.
7. **Enhanced Decision Support**
   - **Benefit**: Provides actionable insights and evidence-based recommendations to clinicians.
   - **Impact**: Supports better clinical decision-making, improving patient outcomes.

Disadvantages

1. **Data Quality and Availability**
    - **Issue**: High-quality, continuous blood pressure data is often scarce.
    - **Impact**: Limits the accuracy and generalizability of ML models.
2. **Data Privacy and Security**
    - **Issue**: Personal health data collection raises significant privacy and security concerns.
    - **Impact**: Patients may be reluctant to share data, affecting data availability and model training.
3. **Model Interpretability**
    - **Issue**: Complex ML models, especially deep learning, can be difficult to interpret.
    - **Impact**: Clinicians may have difficulty understanding and trusting model predictions.
4. **Integration with Clinical Workflows**
    - **Issue**: Integrating ML models with existing clinical systems can be challenging.
    - **Impact**: Potential disruption to clinical workflows and resistance from healthcare providers.
5. **Generalizability of Models**
    - **Issue**: Models trained on specific datasets may not perform well across different populations or settings.
    - **Impact**: Limits the applicability of ML models to diverse patient groups.
6. **Handling Imbalanced Data**
    - **Issue**: Blood pressure data often contains imbalanced classes, with fewer instances of hypertensive episodes.
    - **Impact**: Biases in the model, leading to poor performance in detecting hypertension.
7. **Dependence on Technology**
    - **Issue**: Reliance on wearable devices and continuous monitoring technologies.
    - **Impact**: Accessibility issues for patients who cannot afford or are not comfortable using such devices.

APPLICATIONS

1. Real-Time Blood Pressure Monitoring

- **Description**: Continuous tracking of blood pressure levels using wearable devices or mobile health applications.
- **Technology**: Wearable sensors, smartwatches, and mobile health apps.
- **Example**: **Omron Heart Guide**: A smartwatch that measures blood pressure and provides real-time data.
- **Benefits**: Provides constant data for immediate health status updates, early detection of abnormal trends, and allows for proactive management of hypertension.

2.predictive analytics for hypertension management

- **Description**: Forecasting future blood pressure trends and potential hypertensive episodes based on historical data.
- **Technology**: Time-series analysis, regression models, ensemble methods.
- **Example**: **Hypertension Prediction Models**: Algorithms that predict future blood pressure levels and hypertensive events based on past patient data.
- **Benefits**: Enables early intervention, reduces the risk of hypertensive crises, and aids in long-term health planning.

3.personalised Treatment plans

- **Description**: Developing customized treatment recommendations based on individual patient data.
- **Technology**: Machine learning algorithms for personalized medicine, recommendation systems.
- **Example**: **Personalized BP Management Systems**: Systems that analyze a patient's health data to recommend personalized medication and lifestyle changes.
- **Benefits**: Tailors treatment to individual needs, improves patient adherence, and enhances treatment effectiveness.

4.Remote patient monitoring

- **Description**: Monitoring patients' blood pressure from a distance using connected devices and
- telehealth platforms.
- **Technology**: IoT devices, telemedicine platforms, cloud-based analytics.
- **Example**: **Withings BPM Connect**: A device that allows patients to measure their
- blood pressure and share the results with their healthcare providers remotely.

- **Benefits**: Reduces the need for in-person visits, facilitates continuous care, and improves accessibility for remote or busy patients.

9.CONCLUSION

In conclusion, machine learning represents a significant advancement in the field of blood pressure analysis, offering numerous benefits such as improved accuracy, proactive management, real-time monitoring, and personalized treatment. These advancements have the potential to transform hypertension care, making it more effective and patient-centered. However, addressing challenges related to data quality, model interpretability, and integration into clinical workflows is essential for the continued success of ML applications in this field.

Looking ahead, the future of ML in blood pressure analysis is filled with opportunities for further innovation and improvement. Continued research and development will drive progress in this area, leading to new applications, enhanced technologies, and better health outcomes for patients managing hypertension.

10.FUTURE SCOPE

1.Advancements in Wearable and Remote Monitoring Technologies

- **Description**: Improving wearable devices and remote monitoring solutions for more accurate, user- friendly, and accessible blood pressure measurement.
- **Technologies**: Advanced sensors, miniaturized devices, real-time data processing.
- **Benefits**: Enhances the reliability of remote blood pressure monitoring, supports continuous health tracking, and increases patient compliance.

2. Enhanced Predictive Analytics for Early Detection and Intervention

- **Description**: Using advanced ML algorithms to predict future blood pressure trends and detect potential health issues before they become serious.
- **Technologies**: Predictive modeling, time-series analysis.
- **Benefits**: Enables early intervention and prevention strategies, helping to manage hypertension proactively.

3. Development of Advanced Anomaly Detection Systems

- **Description**: Creating sophisticated ML systems for detecting unusual blood pressure patterns and predicting potential health emergencies.
- **Technologies**: Anomaly detection algorithms, real-time data analysis.
- **Benefits**: Early identification of potential health issues, leading to timely medical interventions and better patient care.

4. **Improving Data Privacy and Security**

- **Description**: Enhancing methods for safeguarding patient data while using ML technologies for blood pressure analysis.
- **Technologies**: Data encryption, secure data sharing protocols, privacy-preserving machine learning.
- **Benefits**: Ensures patient data security and privacy, fosters trust in ML-based healthcare solutions.

5. Expansion of Telehealth and Remote Monitoring Solutions

- **Description**: Enhancing telehealth platforms and remote monitoring solutions for more effective blood pressure management.

- **Technologies**: Telehealth platforms, remote diagnostics, cloud-based health solutions.
- **Benefits**: Expands access to healthcare services, supports remote patient management, and improves patient engagement.

12.APPENDIX

Model building :

1)Dataset
2)Google colab and VS code Application Building

1. HTML file (Index file, Predict file)
1. CSS file
2. Models in pickle format

SOURCE CODE:

INDEX.HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Predictive Pulse: Harnessing Machine Learning for Blood Pressure Analysis</title>
  <meta content="" name="description">
  <meta content="" name="keywords">

  <!-- Favicons -->
  <link href="/Flask/static/img/favicon.png" rel="icon">
  <link href="/Flask/static/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Roboto:300,300
i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">

  <!-- Vendor CSS Files -->
  <link href="/Flask/static/vendor/fontawesome-free/css/all.min.css" rel="stylesheet">
```

```html
  <link href="/Flask/static/vendor/animate.css/animate.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/aos/aos.css" rel="stylesheet">
  <link href="/Flask/static/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="/Flask/static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

  <!-- Template CSS File -->
  <link href="/Flask/static/css/style.css" rel="stylesheet">

  <!-- =======================================================
  * Template Name: Medicio
  * Updated: Jan 29 2024 with Bootstrap v5.3.2
  * Template URL: https://bootstrapmade.com/medicio-free-bootstrap-theme/
  * Author: BootstrapMade.com
  * License: https://bootstrapmade.com/license/
  ======================================================== -->
</head>

<body>

  <!-- ======= Top Bar ======= -->
  <div id="topbar" class="d-flex align-items-center fixed-top">
    <div class="container d-flex align-items-center justify-content-center justify-content-md-between">
      <div class="align-items-center d-none d-md-flex">
        <i class="bi bi-clock"></i> Monday - Saturday, 8AM to 10PM
      </div>
      <div class="d-flex align-items-center">
        <i class="bi bi-phone"></i> Call us now +1 5589 55488 55
      </div>
    </div>
  </div>

  <!-- ======= Header ======= -->
  <header id="header" class="fixed-top">
    <div class="container d-flex align-items-center">

      <a href="index.html" class="logo me-auto"><img src="/Flask/static/img/logo.png" alt=""></a>
      <!-- Uncomment below if you prefer to use an image logo -->
      <!-- <h1 class="logo me-auto"><a href="index.html">Medicio</a></h1> -->

      <nav id="navbar" class="navbar order-last order-lg-0">
        <ul>
```

```
          <li><a class="nav-link scrollto " href="/">Home</a></li>
          <li><a class="nav-link scrollto" href="#about">About</a></li>
          <li><a class="nav-link scrollto" href="#footer">Contact</a></li>
        </ul>
        <i class="bi bi-list mobile-nav-toggle"></i>
      </nav><!-- .navbar -->

      <a href="#appointment" class="appointment-btn scrollto"><span class="d-none d-md-inline">Check
your</span> Blood Pressure</a>

    </div>
  </header><!-- End Header -->

  <!-- ======= Hero Section ======= -->
  <section id="hero">
    <div id="heroCarousel" data-bs-interval="5000" class="carousel slide carousel-fade" data-bs-ride="carousel">

      <ol class="carousel-indicators" id="hero-carousel-indicators"></ol>

      <div class="carousel-inner" role="listbox">

        <!-- Slide 1 -->
        <div class="carousel-item active" style="background-image: url(/Flask/static/img/slide/slide-1.jpg)">
          <div class="container">
            <h2>Welcome to <span>Medicio</span></h2>
            <p>Welcome to our platform dedicated to monitoring blood pressure stages. Discover your current status
and take proactive steps towards better health management.</p>
            <a href="#about" class="btn-get-started scrollto">Read More</a>
          </div>
        </div>

      </div>

    </div>
  </section><!-- End Hero -->

  <main id="main">


    <!-- ======= About Us Section ======= -->
    <section id="about" class="about">
      <div class="container" data-aos="fade-up">

        <div class="section-title">
```

```
        <h2>Information.</h2>
        <p> Blood pressure (BP) measures the force of blood pushing against artery walls.
        Maintaining healthy BP levels is crucial for overall cardiovascular health and preventing diseases like heart
attack and stroke.</p>
      </div>

      <div class="row">
        <div class="col-lg-6" data-aos="fade-right">
          <img src="/Flask/static/img/about.jpg" class="img-fluid" alt="">
        </div>
        <div class="col-lg-6 pt-4 pt-lg-0 content" data-aos="fade-left">
          <h3>Stages</h3>
          <ul>
            <li><i class="bi bi-check-circle"></i>Stage 1: Normal: Systolic < 120 mm Hg, Diastolic < 80 mm Hg.</li>
            <li><i class="bi bi-check-circle"></i>Stage 2: Elevated: Systolic 120-129 mm Hg, Diastolic < 80 mm Hg. </li>
            <li><i class="bi bi-check-circle"></i>Stage 3: Hypertension Stage 1: Systolic 130-139 mm Hg, Diastolic 80-
89 mm Hg.</li>
            <li><i class="bi bi-check-circle"></i>Stage 4: Hypertension Stage 2: Systolic ≥ 140 mm Hg, Diastolic ≥ 90
mm Hg.</li>
          </ul>
          <p>
            <h3>Risk Factors:</h3>
              Family history increases risk.High salt, low potassium diets.Lack of physical activity, excessive alcohol
consumption.
              Increases strain on heart and blood vessels.Long-term stress raises BP.Risk increases with age.
          </p>
        </div>
      </div>

    </div>
  </section><!-- End About Us Section -->


  <!-- ======= Appointment Section ======= -->
  <section id="appointment" class="appointment section-bg">
    <div class="container" data-aos="fade-up">

      <div class="section-title">
        <h2>Check your Health!</h2>
        <p>Checking blood pressure stage is crucial to assess cardiovascular health, detect hypertension or
hypotension, and mitigate risks of heart disease, stroke, or organ damage, ensuring timely intervention.</p>
      </div>
<!--forms/appointment.php-->
```

```
<!--<form action="{{ url_for('predict')}}" method="post" role="form" class="php-email-form" data-aos="fade-
up" data-aos-delay="100">-->
<form action="{{ url_for('predict')}}" method="post" >
  <div class="row">
    <div class="col-md-4 form-group">
      <select name="Gender" id="Gender" class="form-select">
        <option value=>Select Gender</option>
        <option value=1>Male</option>
        <option value=0>Female</option>
      </select>
    </div>
    <div class="col-md-4 form-group mt-3 mt-md-0">
      <select name="Age" id="Age" class="form-select">
        <option value=>Select Age</option>
        <option value=0>18-34</option>
        <option value=1>35-50</option>
        <option value=3>51-64</option>
        <option value=4>65+</option>
      </select>
    </div>
    <div class="col-md-4 form-group mt-3 mt-md-0">
      <select name="Patient" id="Patient" class="form-select">
        <option value=>Select Patient</option>
        <option value=1>Yes</option>
        <option value=0>No</option>
      </select>
    </div>
  </div>
  <div class="row">
    <div class="col-md-4 form-group mt-3">
      <select name="Severity" id="Severity" class="form-select">
        <option value=>Select Severity</option>
        <option value=1>Yes</option>
        <option value=0>No</option>
      </select>
    </div>
    <div class="col-md-4 form-group mt-3">
      <select name="BreathShortness" id="BreathShortness" class="form-select">
        <option value=>Do you have Breath Shortness</option>
        <option value=1>Yes</option>
        <option value=0>No</option>
      </select>
    </div>
    <div class="col-md-4 form-group mt-3">
```

```
      <select name="VisualChanges" id="VisualChanges" class="form-select">
        <option value=>Are there are VisualChanges</option>
        <option value=1>Yes</option>
        <option value=0>No</option>
      </select>
    </div>
  </div>
  <div class="row">
    <div class="col-md-4 form-group mt-3">
      <select name="NoseBleeding" id="NoseBleeding" class="form-select">
        <option value=>Do you have nose bleeding</option>
        <option value=1>Yes</option>
        <option value=0>No</option>
      </select>
    </div>
    <div class="col-md-4 form-group mt-3">
      <select name="Whendiagnoused" id="Whendiagnoused" class="form-select">
        <option value=>When diagnoused</option>
        <option value=0> less than 1 year</option>
        <option value=3> 1- 5 years</option>
        <option value=6> greater than 5 years</option>
      </select>
    </div>
    <div class="col-md-4 form-group mt-3">
      <select name="Systolic" id="Systolic" class="form-select">
        <option value=>Select Systolic</option>
        <option value=115>100+</option>
        <option value=125>111-120</option>
        <option value=135>121-130</option>
        <option value=145>130+</option>
      </select>
    </div>
  </div>
  <div class="row">
    <div class="col-md-4 form-group mt-3">
      <select name="Diastolic" id="Diastolic" class="form-select">
        <option value=>Select Diastolic</option>
        <option value=75>70-80</option>
        <option value=85>81-90</option>
        <option value=95>91-100</option>
        <option value=115>100+</option>
      </select>
    </div>
    <div class="col-md-4 form-group mt-3">
```

```
        <select name="ControlledDiet" id="ControlledDiet" class="form-select">
          <option value=>Any Diet Followed</option>
          <option value=1>Yes</option>
          <option value=0>No</option>
        </select>
      </div>
    </div>
    <div class="text-center">
      <button type="submit">Submit</button>
    </div>

  </form>
</section><!-- End Appointment Section -->

</main><!-- End #main -->

<!-- ======= Footer ======= -->
<footer id="footer">
 <div class="footer-top">
   <div class="container">
     <div class="row">

       <div class="col-lg-3 col-md-6">
         <div class="footer-info">
           <h3>Medicio</h3>
           <p>
             Hyderabad, Telangana,India<br><br>
             <strong>Phone:</strong> +1 5589 55488 55<br>
             <strong>Email:</strong> info@example.com<br>
           </p>
           <div class="social-links mt-3">
             <a href="#" class="twitter"><i class="bx bxl-twitter"></i></a>
             <a href="#" class="facebook"><i class="bx bxl-facebook"></i></a>
             <a href="#" class="instagram"><i class="bx bxl-instagram"></i></a>
             <a href="#" class="google-plus"><i class="bx bxl-skype"></i></a>
             <a href="#" class="linkedin"><i class="bx bxl-linkedin"></i></a>
           </div>
         </div>
       </div>

     </div>
   </div>
 </div>
```

```
  <!-- End Footer -->

  <div id="preloader"></div>
  <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-
short"></i></a>

  <!-- Vendor JS Files -->
  <script src="/Flask/static/vendor/purecounter/purecounter_vanilla.js"></script>
  <script src="/Flask/static/vendor/aos/aos.js"></script>
  <script src="/Flask/static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="/Flask/static/vendor/glightbox/js/glightbox.min.js"></script>
  <script src="/Flask/static/vendor/swiper/swiper-bundle.min.js"></script>
  <script src="/Flask/static/vendor/php-email-form/validate.js"></script>

  <!-- Template JSFile -->
  <script src="/Flask/static/js/main.js"></script>

</body>

</html>
```

PREDICT.HTML

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <meta content="width=device-width, initial-scale=1.0" name="viewport">

  <title>Predictive Pulse: Harnessing Machine Learning for Blood Pressure Analysis</title>
  <meta content="" name="description">
  <meta content="" name="keywords">



  <!-- Favicons -->
  <link href="/Flask/static/img/favicon.png" rel="icon">
  <link href="/Flask/static/img/apple-touch-icon.png" rel="apple-touch-icon">

  <!-- Google Fonts -->
  <link
href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,700,700i|Roboto:300,300
i,400,400i,500,500i,600,600i,700,700i|Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
```

```
  <!-- Vendor CSS Files -->
  <link href="/Flask/static/vendor/fontawesome-free/css/all.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/animate.css/animate.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/aos/aos.css" rel="stylesheet">
  <link href="/Flask/static/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
  <link href="/Flask/static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/glightbox/css/glightbox.min.css" rel="stylesheet">
  <link href="/Flask/static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">

  <!-- Template CSS File -->
  <link href="/Flask/static/css/style.css" rel="stylesheet">

  <!-- =======================================================
  * Template Name: Medicio
  * Updated: Jan 29 2024 with Bootstrap v5.3.2
  * Template URL: https://bootstrapmade.com/medicio-free-bootstrap-theme/
  * Author: BootstrapMade.com
  * License: https://bootstrapmade.com/license/
  ======================================================== -->
</head>

<body>

  <!-- ======= Top Bar ======= -->
  <div id="topbar" class="d-flex align-items-center fixed-top">
    <div class="container d-flex align-items-center justify-content-center justify-content-md-between">
      <div class="align-items-center d-none d-md-flex">
        <i class="bi bi-clock"></i> Monday - Saturday, 8AM to 10PM
      </div>
      <div class="d-flex align-items-center">
        <i class="bi bi-phone"></i> Call us now +1 5589 55488 55
      </div>
    </div>
  </div>

  <!-- ======= Header ======= -->
  <header id="header" class="fixed-top">
    <div class="container d-flex align-items-center">

      <a href="index.html" class="logo me-auto"><img src="/Flask/static/img/logo.png" alt=""></a>
      <!-- Uncomment below if you prefer to use an image logo -->
```

```
    <!-- <h1 class="logo me-auto"><a href="index.html">Medicio</a></h1> -->

    <nav id="navbar" class="navbar order-last order-lg-0">
      <ul>
        <li><a class="" href="/">Home</a></li>
        <!--<li><a class="nav-link scrollto" href="#about"></a></li>-->

        <!--<li><a class="nav-link scrollto" href="#footer">Contact</a></li>-->

      </ul>
      <i class="bi bi-list mobile-nav-toggle"></i>
    </nav><!-- .navbar -->

    <a href="/Flask/templates/index.html#appointment" class="appointment-btn scrollto"><span class="d-none
d-md-inline">Check your</span> Blood Pressure</a>

   </div>
  </header><!-- End Header -->

  <main id="main">

   <!-- ======= Breadcrumbs Section ======= -->
   <section class="breadcrumbs">
    <div class="container">

      <div class="d-flex justify-content-between align-items-center">
        <h2>Result Page</h2>
        <ol>
          <li><a href="index.html">Home</a></li>
        </ol>
      </div>

    </div>
   </section><!-- End Breadcrumbs Section -->

   <section class="inner-page">
    <div class="container">
      <form id="appointment data-form" onsubmit="return calculatebloodpressure()">
        <h1>{{ prediction_text }}</h1>
    </form>
    </div>
   </section>
```

```
  </main><!-- End #main -->

  <!-- ======= Footer ======= -->
  <footer id="footer">
    <div class="footer-top">
      <div class="container">
        <div class="row">

          <div class="col-lg-3 col-md-6">
            <div class="footer-info">
              <h3>Medicio</h3>
              <p>
                The Smart Bridge<br>
                Hyderabad, Telangana,India<br><br>
                <strong>Phone:</strong> +1 5589 55488 55<br>
                <strong>Email:</strong> info@example.com<br>
              </p>
              <div class="social-links mt-3">
                <a href="#" class="twitter"><i class="bx bxl-twitter"></i></a>
                <a href="#" class="facebook"><i class="bx bxl-facebook"></i></a>
                <a href="#" class="instagram"><i class="bx bxl-instagram"></i></a>
                <a href="#" class="google-plus"><i class="bx bxl-skype"></i></a>
                <a href="#" class="linkedin"><i class="bx bxl-linkedin"></i></a>
              </div>
            </div>
          </div>


        </div>
      </div>
    </div>

    <!-- End Footer -->

  <div id="preloader"></div>
  <a href="#" class="back-to-top d-flex align-items-center justify-content-center"><i class="bi bi-arrow-up-
short"></i></a>

  <!-- Vendor JS Files -->
  <script src="/Flask/static/vendor/purecounter/purecounter_vanilla.js"></script>
  <script src="/Flask/static/vendor/aos/aos.js"></script>
```

```
  <script src="/Flask/static/vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
  <script src="/Flask/static/vendor/glightbox/js/glightbox.min.js"></script>
  <script src="/Flask/static/vendor/swiper/swiper-bundle.min.js"></script>
  <script src="/Flask/static/vendor/php-email-form/validate.js"></script>

  <!-- Template Main JS File -->
  <script src="/Flask/static/js/main.js"></script>
</body>

</html>
```

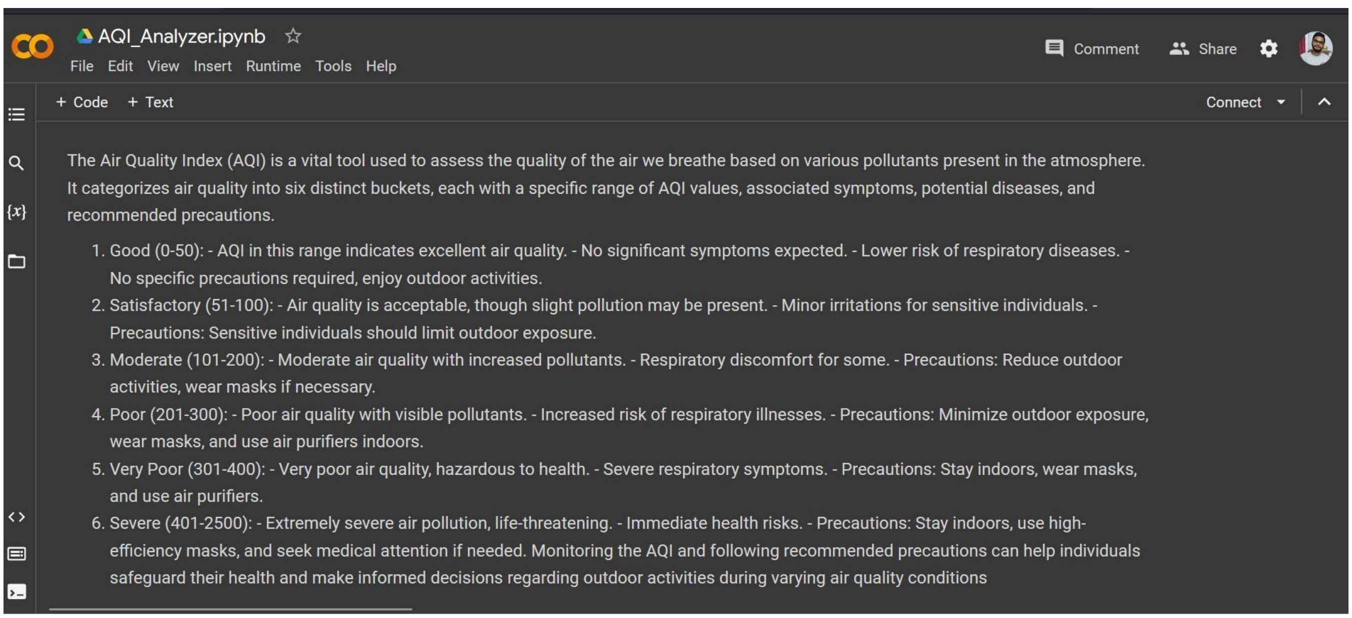
APP.PY

```
import pickle
import pandas as pd
import numpy as np
from flask import Flask, request, render_template, redirect, url_for

app = Flask(__name__,static_url_path='/Flask/static')
model = pickle.load(open('model.pkl','rb'))

@app.route('/')
def home():
   return render_template('index.html')

@app.route('/predict', methods=["POST"])
def predict():
   Gender = float(request.form["Gender"])

   Age = float(request.form["Age"])
   Patient = float(request.form['Patient'])
   Severity = float(request.form['Severity'])
   BreathShortness = float(request.form['BreathShortness'])
   VisualChange = float(request.form['VisualChanges'])
   NoseBleeding = float(request.form['NoseBleeding'])
   Whendiagnoused = float(request.form['Whendiagnoused'])
   Systolic = float(request.form['Systolic'])
   Diastolic = float(request.form['Diastolic'])
   ControlledDiet = float(request.form['ControlledDiet'])

   features_values=np.array([[Gender,Age,Patient,Severity,BreathShortness,VisualChange,NoseBleeding,
Whendiagnoused,Systolic,Diastolic,ControlledDiet]])



   df = pd.DataFrame(features_values, columns=['Gender','Age','Patient','Severity','BreathShortness','VisualChanges',
                     'NoseBleeding','Whendiagnoused','Systolic','Diastolic','ControlledDiet'])
   print(df)

   prediction = model.predict(df)
   print(prediction[0])
```

```
    if prediction[0] == 0:
      result="NORMAL"
    elif prediction[0] == 1:
      result="HYPERTENSION (Stage-1)"
    elif prediction[0] == 2:
      result='HYPERTENSION (Stage-2)'
    else:
      result='HYPERTENSIVE CRISIS'
    print(result)
    text = "Your Blood Pressure stage is: "
    return render_template("predict.html", prediction_text=text + result)


if __name__ == "__main__":
    app.run(debug=False, port=5000)
```

CODE SNIPPETS

MODEL BUILDING



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
#Importing data
df = pd.read_csv('patient_data.csv')
```

```
df.head()
```

	C	Age	History	Patient	TakeMedication	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnoused	Systolic	Diastolic	ControlledDiet
0	Male	18-34	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
1	Female	18-34	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
2	Male	35-50	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
3	Female	35-50	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
4	Male	51-64	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No

```
df.rename(columns={'C': 'Gender'}, inplace=True)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1825 entries, 0 to 1824
Data columns (total 14 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Gender           1825 non-null   object
 1   Age              1825 non-null   object
 2   History          1825 non-null   object
 3   Patient          1825 non-null   object
 4   TakeMedication   1825 non-null   object
 5   Severity         1825 non-null   object
 6   BreathShortness  1825 non-null   object
 7   VisualChanges    1825 non-null   object
 8   NoseBleeding     1825 non-null   object
 9   Whendiagnoused   1825 non-null   object
 10  Systolic         1825 non-null   object
 11  Diastolic        1825 non-null   object
 12  ControlledDiet   1825 non-null   object
 13  Stages           1825 non-null   object
dtypes: object(14)
memory usage: 199.7+ KB
```

```
df.shape
```

```
(1825, 14)
```

```
df.describe()
```

	Gender	Age	History	Patient	TakeMedication	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnoused	Systolic	Diastolic	Controlle
count	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825	1825	
unique	2	4	2	2	3	3	2	2	3	3	5	5	
top	Female	51-64	Yes	No	No	Moderate	No	No	No	<1 Year	111 - 120	81 - 90	
freq	913	475	1657	984	744	697	976	940	984	625	1008	708	

Data Preprocessing

```
#checking for null values
df.isnull().sum()
```

[8] Python

```
Gender            0
Age               0
History           0
Patient           0
TakeMedication    0
Severity          0
BreathShortness   0
VisualChanges     0
NoseBleeding      0
Whendiagnoused    0
Systolic          0
Diastolic         0
ControlledDiet    0
Stages            0
dtype: int64
```

```
df['Stages'].unique()
```

[9] Python

```
array(['HYPERTENSION (Stage-1)', 'HYPERTENSION (Stage-2)',
       'HYPERTENSIVE CRISIS', 'HYPERTENSION (Stage-2).',
       'HYPERTENSIVE CRISI', 'NORMAL'], dtype=object)
```

```
df['Stages'].replace({'HYPERTENSIVE CRISI': 'HYPERTENSIVE CRISIS',
                      'HYPERTENSION (Stage-2).': 'HYPERTENSION (Stage-2)'},
                     inplace=True)
```

[10] Python

```
C:\Users\suchi\AppData\Local\Temp\ipykernel_13988\768679657.py:1: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through c
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) inste


  df['Stages'].replace({'HYPERTENSIVE CRISI': 'HYPERTENSIVE CRISIS',
```

```
df['Stages'].unique()
```

[11] Python

```
array(['HYPERTENSION (Stage-1)', 'HYPERTENSION (Stage-2)',
       'HYPERTENSIVE CRISIS', 'NORMAL'], dtype=object)
```

```
df.head()
```

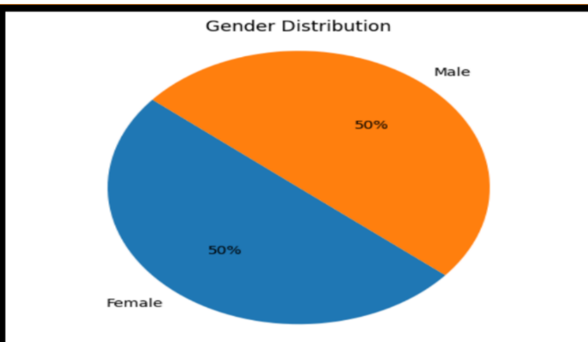
[12] Python

	Gender	Age	History	Patient	TakeMedication	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnoused	Systolic	Diastolic	ControlledDiet
0	Male	18-34	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
1	Female	18-34	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
2	Male	35-50	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
3	Female	35-50	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No
4	Male	51-64	Yes	No	No	Mild	No	No	No	<1 Year	111 - 120	81 - 90	No

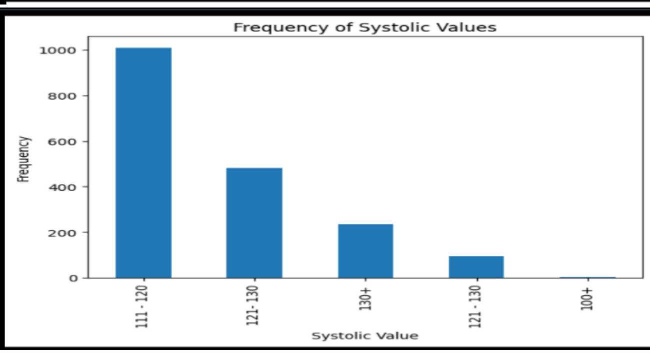
```python
df['Whendiagnoused'].unique()
```
[13] Python

```
array(['<1 Year', '1 - 5 Years', '>5 Years'], dtype=object)
```

```python
gender_counts = df['Gender'].value_counts()

# Plotting the pie chart
plt.pie(gender_counts, labels=gender_counts.index, autopct='%1.0f%%', startangle=140)
plt.title('Gender Distribution')
plt.axis('equal')
plt.show()
```
[14] Python

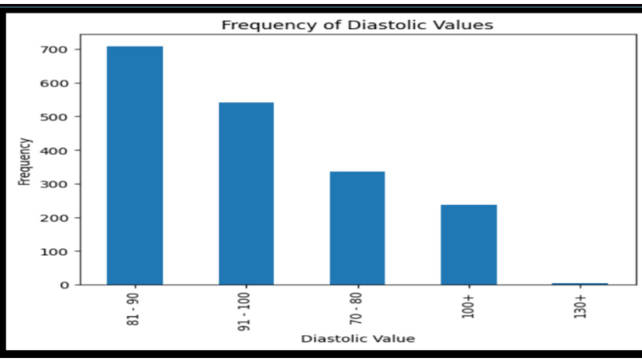



```python
frequency = df['Systolic'].value_counts()

frequency.plot(kind='bar')
plt.xlabel('Systolic Value')
plt.ylabel('Frequency')
plt.title('Frequency of Systolic Values')
plt.show()
```
[15] Python



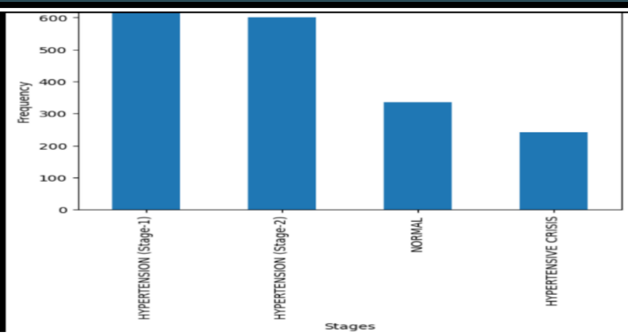

```python
frequency = df['Diastolic'].value_counts()

frequency.plot(kind='bar')
plt.xlabel('Diastolic Value')
plt.ylabel('Frequency')
plt.title('Frequency of Diastolic Values')
plt.show()
```
[16] Python

```
...
```




```
frequency = df['Stages'].value_counts()

frequency.plot(kind='bar')
plt.xlabel('Stages')
plt.ylabel('Frequency')
plt.title('Count of stages')
plt.show()
```

[17] Python




```
df['Whendiagnoused'].unique()
```

[18] Python

```
array(['<1 Year', '1 - 5 Years', '>5 Years'], dtype=object)
```

```
df['Age'].unique()
```

[19] Python

```
array(['18-34', '35-50', '51-64', '65+'], dtype=object)
```

```
#converting categorical into numerical value
from sklearn.preprocessing import LabelEncoder

columns = ['Gender' ,'Severity' ,'History' , 'Patient','TakeMedication', 'BreathShortness',
           'VisualChanges','NoseBleeding','ControlledDiet','Stages']

label_encoder = LabelEncoder()
for col in columns:
    df[col] = label_encoder.fit_transform(df[col])
```

[20] Python

```python
df
#(1 = Male, 0 = Female)(1 = yes, 0 = no)
```
[21] Python

+ Code + Markdown ··· Select Kernel

	Gender	Age	History	Patient	TakeMedication	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnoused	Systolic	Diastolic	ControlledDi
0	1	18-34	1	0	0	0	0	0	1	<1 Year	111 - 120	81 - 90	
1	0	18-34	1	0	0	0	0	0	1	<1 Year	111 - 120	81 - 90	
2	1	35-50	1	0	0	0	0	0	1	<1 Year	111 - 120	81 - 90	
3	0	35-50	1	0	0	0	0	0	1	<1 Year	111 - 120	81 - 90	
4	1	51-64	1	0	0	0	0	0	1	<1 Year	111 - 120	81 - 90	
...	...	...	...	...	...	...	...	...	...	...	...	...	
1820	0	35-50	1	0	0	2	0	0	1	>5 Years	111 - 120	70 - 80	
1821	1	51-64	1	0	0	2	0	0	1	>5 Years	111 - 120	70 - 80	
1822	0	51-64	1	0	0	2	0	0	1	>5 Years	111 - 120	70 - 80	
1823	1	65+	1	0	0	2	0	0	1	>5 Years	111 - 120	70 - 80	
1824	0	65+	1	0	0	2	0	0	1	>5 Years	111 - 120	70 - 80	

1825 rows × 14 columns

```python
# Replace '+' with a large number in 'Systolic' and 'Diastolic' columns
df['Systolic'] = df['Systolic'].astype(str).str.replace('+', '150')
df['Diastolic'] = df['Diastolic'].astype(str).str.replace('+', '150')

# Convert range values to the average in 'Systolic' and 'Diastolic' columns
df['Systolic'] = df['Systolic'].apply(lambda x: int((int(x.split('-')[0].strip()) + int(x.split('-')[1].strip())) / 2) if '-' in x else int(x))
df['Diastolic'] = df['Diastolic'].apply(lambda x: int((int(x.split('-')[0].strip()) + int(x.split('-')[1].strip())) / 2) if '-' in x else int(x))

# Convert categorical values to numerical values in 'Whendiagnoused' column
df['Whendiagnoused'] = df['Whendiagnoused'].map({'<1 Year': 0, '1 - 5 Years': 3, '>5 Years': 6})
df['Age'] = df['Age'].map({'18-34': 0, '35-50': 1,'51-64': 2, '65+': 3})
```
[22] Python

```python
df.head()
```
[23] Python

	Gender	Age	History	Patient	TakeMedication	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnoused	Systolic	Diastolic	ControlledDiet
0	1	0	1	0	0	0	0	0	1	0	115	85	0
1	0	0	1	0	0	0	0	0	1	0	115	85	0
2	1	1	1	0	0	0	0	0	1	0	115	85	0
3	0	1	1	0	0	0	0	0	1	0	115	85	0
4	1	2	1	0	0	0	0	0	1	0	115	85	0

```python
# Scaling numerical features using Min-Max scaling

from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()
df[['Systolic', 'Diastolic']] = scaler.fit_transform(df[['Systolic', 'Diastolic']])
```
[24] Python

+ Code + Markdown

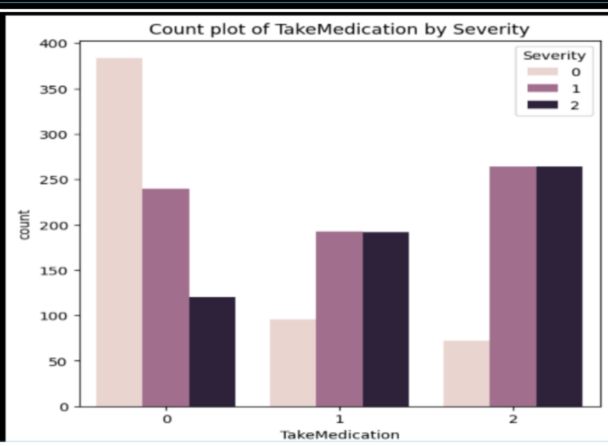
```python
# Compute the correlation matrix
correlation_matrix = df.corr()

# Print or visualize the correlation matrix
print(correlation_matrix)
```
[25] Python

```
                 Gender       Age   History   Patient  TakeMedication  \
Gender         1.000000 -0.000279 -0.000174 -0.000593       -0.000051
Age           -0.000279  1.000000 -0.017075 -0.058006       -0.067999
History       -0.000174 -0.017075  1.000000  0.294370        0.343221
Patient       -0.000593 -0.058006  0.294370  1.000000        0.857211
TakeMedication -0.000051 -0.067999  0.343221  0.857211        1.000000
Severity       0.000009  0.000897  0.121062  0.420845        0.373521
BreathShortness -0.000588 -0.045496  0.296977  0.797282       0.682626
VisualChanges  -0.000565 -0.061256  0.308959  0.838404        0.731114
NoseBleeding   -0.000786  0.090742  0.012604  0.042816       -0.177849
Whendiagnoused  0.000662 -0.043753 -0.055196 -0.064498       -0.074472
Systolic        0.000213  0.072117  0.123868  0.185770       -0.057170
Diastolic      -0.001396  0.076996  0.124202  0.189244       -0.052026
ControlledDiet -0.000593 -0.058006  0.294370  1.000000        0.857211
Stages         -0.000427  0.007328 -0.538581  0.119174       -0.142148
```

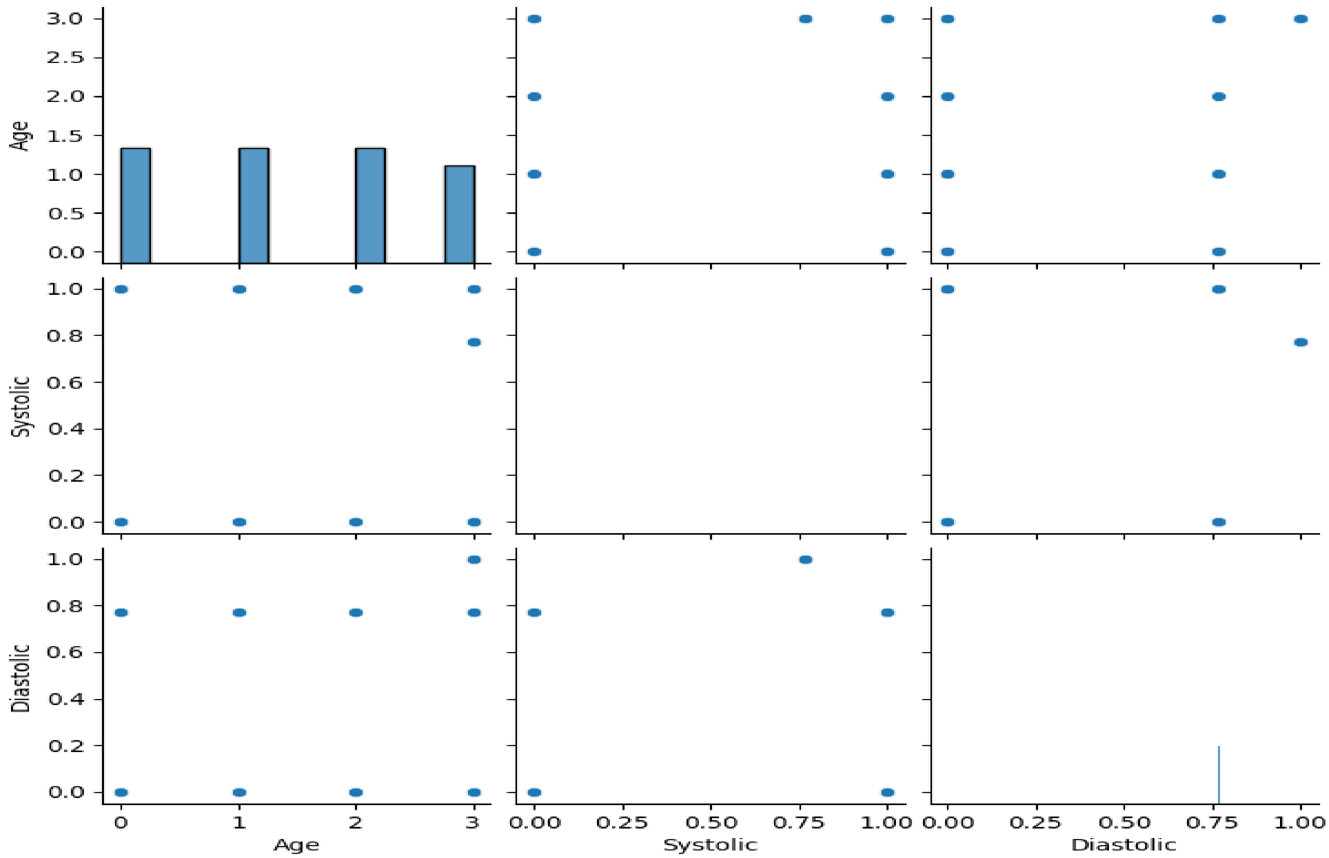
```
               Severity  BreathShortness  VisualChanges  NoseBleeding  \
Gender         0.000009        -0.000588      -0.000565     -0.000786
Age            0.000897        -0.045496      -0.061256      0.090742
History        0.121062         0.296977       0.308959      0.012604
Patient        0.420845         0.797282       0.838404      0.042816
TakeMedication 0.373521         0.682626       0.731114     -0.177849
Severity       1.000000         0.426014       0.474726      0.147320
BreathShortness 0.426014        1.000000       0.715029      0.146091
VisualChanges  0.474726         0.715029       1.000000      0.124845
...
Systolic             -0.045128  1.000000   0.307082        0.185770  0.055297
Diastolic            -0.050717  0.307082   1.000000        0.189244  0.055402
ControlledDiet       -0.064498  0.185770   0.189244        1.000000  0.119174
Stages                0.057469  0.055297   0.055402        0.119174  1.000000
```

Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...

```python
# Count plot of TakeMedication and Severity
plt.figure(figsize=(6, 6))
sns.countplot(x='TakeMedication', hue='Severity', data=df)
plt.title('Count plot of TakeMedication by Severity')
plt.show()
```




```python
sns.pairplot(df[['Age', 'Systolic', 'Diastolic']])
plt.show()
```




```python
df.drop(columns=['History', 'TakeMedication'], inplace=True)
```

[28] Python

```python
df
```

[29] Python

	Gender	Age	Patient	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnoused	Systolic	Diastolic	ControlledDiet	Stages
0	1	0	0	0	0	0	1	0	0.0	0.000077	0	0
1	0	0	0	0	0	0	1	0	0.0	0.000077	0	0
2	1	1	0	0	0	0	1	0	0.0	0.000077	0	0
3	0	1	0	0	0	0	1	0	0.0	0.000077	0	0
4	1	2	0	0	0	0	1	0	0.0	0.000077	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
1820	0	1	0	2	0	0	1	6	0.0	0.000000	0	3
1821	1	2	0	2	0	0	1	6	0.0	0.000000	0	3
1822	0	2	0	2	0	0	1	6	0.0	0.000000	0	3
1823	1	3	0	2	0	0	1	6	0.0	0.000000	0	3
1824	0	3	0	2	0	0	1	6	0.0	0.000000	0	3

1825 rows × 12 columns

```
#Splitting the data into X and Y
X = df.drop('Stages' , axis = 1)
X
```

[30]

	Gender	Age	Patient	Severity	BreathShortness	VisualChanges	NoseBleeding	Whendiagnoused	Systolic	Diastolic	ControlledDiet
0	1	0	0	0	0	0	1	0	0.0	0.000077	0
1	0	0	0	0	0	0	1	0	0.0	0.000077	0
2	1	1	0	0	0	0	1	0	0.0	0.000077	0
3	0	1	0	0	0	0	1	0	0.0	0.000077	0
4	1	2	0	0	0	0	1	0	0.0	0.000077	0
...	...	...	...	...	...	...	...	...	...	...	...
1820	0	1	0	2	0	0	1	6	0.0	0.000000	0
1821	1	2	0	2	0	0	1	6	0.0	0.000000	0
1822	0	2	0	2	0	0	1	6	0.0	0.000000	0
1823	1	3	0	2	0	0	1	6	0.0	0.000000	0
1824	0	3	0	2	0	0	1	6	0.0	0.000000	0

1825 rows × 11 columns

```
Y = df['Stages']
Y
```

[31] Python

```
0       0
1       0
2       0
3       0
4       0
       ..
1820    3
1821    3
1822    3
1823    3
1824    3
Name: Stages, Length: 1825, dtype: int32
```

Splitting into Train and Test data

```
#splitting into training and testing dataset
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X, Y, test_size=0.2,random_state=30)
```

[32] Python

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

[33] Python

```
(1460, 11)
(365, 11)
(1460,)
(365,)
```

```python
import numpy as np
import pandas as pd
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

# Generate sample data (or load your dataset)
# Example: Using sklearn's breast cancer dataset
from sklearn.datasets import load_breast_cancer
data = load_breast_cancer()
X, y = data.data, data.target

# Split the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Check the shapes of the data
print("Training data shape:", x_train.shape)
print("Training labels shape:", y_train.shape)
print("Testing data shape:", x_test.shape)
print("Testing labels shape:", y_test.shape)

# Standardize the features (optional but recommended for logistic regression)
scaler = StandardScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)
```

```python
# Check for missing values
print("Missing values in training data:", np.sum(np.isnan(x_train)))
print("Missing values in testing data:", np.sum(np.isnan(x_test)))

# Initialize and fit the Logistic Regression model
logistic_regression = LogisticRegression()
logistic_regression.fit(x_train, y_train)

# Make predictions on the test set
y_pred = logistic_regression.predict(x_test)

# Calculate accuracy
acc_lr = accuracy_score(y_test, y_pred)
print("Accuracy:", acc_lr)

# Generate classification report
c_lr = classification_report(y_test, y_pred)
print("Classification Report:\n", c_lr)
```

```
Training data shape: (455, 30)
Training labels shape: (455,)
Testing data shape: (114, 30)
Testing labels shape: (114,)
Missing values in training data: 0
Missing values in testing data: 0
Accuracy: 0.9736842105263158
Classification Report:
               precision    recall  f1-score   support

           0       0.98      0.95      0.96        43
           1       0.97      0.99      0.98        71

    accuracy                           0.97       114
   macro avg       0.97      0.97      0.97       114
weighted avg       0.97      0.97      0.97       114
```

```python
from sklearn.ensemble import RandomForestClassifier
random_forest=RandomForestClassifier()
random_forest.fit(x_train,y_train)
y_pred=random_forest.predict(x_test)
acc_rf=accuracy_score(y_test,y_pred)
c_rf=classification_report(y_test,y_pred)
print("Accuracy Score:",acc_rf)
print(c_rf)
```

Python

```
... Accuracy Score: 0.9649122807017544
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114
```

```python
from sklearn.tree import DecisionTreeClassifier

decision_tree_model= DecisionTreeClassifier()
decision_tree_model.fit(x_train,y_train)
y_pred = decision_tree_model.predict(x_test)

acc_dt= accuracy_score(y_test,y_pred)
c_dt=classification_report(y_test,y_pred)

print("Accuracy Score: ",acc_dt)
print(c_dt)
```

Python

```
... Accuracy Score:  0.9298245614035088
              precision    recall  f1-score   support

           0       0.89      0.93      0.91        43
           1       0.96      0.93      0.94        71

    accuracy                           0.93       114
   macro avg       0.92      0.93      0.93       114
weighted avg       0.93      0.93      0.93       114
```

```python
from sklearn.naive_bayes import GaussianNB # Fixed the typo in module name from 'skleran' to 'sklearn'

NB=GaussianNB()
NB.fit(x_train,y_train)
y_pred=NB.predict(x_test)

acc_nb=accuracy_score(y_test,y_pred)
c_nb=classification_report(y_test,y_pred)

print("Accuracy Score:",acc_nb)
print(c_nb)
```

Python

```
... Accuracy Score: 0.9649122807017544
              precision    recall  f1-score   support

           0       0.98      0.93      0.95        43
           1       0.96      0.99      0.97        71

    accuracy                           0.96       114
   macro avg       0.97      0.96      0.96       114
weighted avg       0.97      0.96      0.96       114
```

```python
from sklearn.ensemble import RandomForestClassifier
import pickle
import warnings

# Assuming x_train and y_train are defined somewhere


pickle.dump(random_forest, open("model.pkl", "wb"))  # Now you can pickle it
```

Pytho

```python
import pickle
import warnings
pickle.dump(random_forest,open("model.pkl","wb"))
```

Pytho

```python
import numpy as np
import pandas as pd
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler, MinMaxScaler
```

```python
# Generate sample data (or load your dataset)
# Example: Using sklearn's breast cancer dataset
#from sklearn.datasets import load_breast_cancer
#data = load_breast_cancer()
#X, y = data.data, data.target

# Split the dataset into training and testing sets
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Check the shapes of the data
print("Training data shape:", x_train.shape)
print("Training labels shape:", y_train.shape)
print("Testing data shape:", x_test.shape)
print("Testing labels shape:", y_test.shape)

# Use MinMaxScaler to ensure non-negative feature values for MultinomialNB
scaler = MinMaxScaler()
x_train = scaler.fit_transform(x_train)
x_test = scaler.transform(x_test)

# Check for missing values
print("Missing values in training data:", np.sum(np.isnan(x_train)))
print("Missing values in testing data:", np.sum(np.isnan(x_test)))

# Initialize and fit the Multinomial Naive Bayes model
mNB = MultinomialNB()
mNB.fit(x_train, y_train)
```

```python
# Make predictions on the test set
y_pred = mNB.predict(x_test)

# Calculate accuracy
acc_mnb = accuracy_score(y_test, y_pred)
print("Accuracy:", acc_mnb)

# Generate classification report
c_mnb = classification_report(y_test, y_pred)
print("Classification Report:\n", c_mnb)
```

```
Training data shape: (455, 30)
Training labels shape: (455,)
Testing data shape: (114, 30)
Testing labels shape: (114,)
Missing values in training data: 0
Missing values in testing data: 0
Accuracy: 0.8508771929824561
Classification Report:
               precision    recall  f1-score   support

           0       1.00      0.60      0.75        43
           1       0.81      1.00      0.89        71

    accuracy                           0.85       114
   macro avg       0.90      0.80      0.82       114
weighted avg       0.88      0.85      0.84       114
```

```python
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import make_classification # Sample data for demonstration

# Create some sample data (replace with your actual data)
X, y = make_classification(n_samples=1000, n_features=13, random_state=42)

# Initialize and fit the RandomForestClassifier
random_forest = RandomForestClassifier(random_state=42)
random_forest.fit(X, y) # Fit the model to the data

# Now you can make predictions
prediction = random_forest.predict([[0,3,0,2,0,0,1,6,0,0,0,0,0]])
print(prediction)
```

```python
prediction[0]
```

Python

```
1
```

```python
if prediction[0] == 0:
    print("NORMAL")
elif prediction[0] == 1:
    print("HYPERTENSION(Stage-1)")
elif prediction[0] == 2:
    print("HYPERTENSION(Stage-2)")
else:
    print("HYPERTENSION CRISIS")
```

```
HYPERTENSION(Stage-1)
```

```python
model=pd.DataFrame({"Model":["Linear Regression","Decision Tree Classifier ",
                             "RandomForest Classifier","Gaussian Navie Bayes","Multinomial Navie Bayes"],
                    "Score":[acc_lr,acc_dt,acc_rf,acc_nb,acc_mnb]})
```
Python

```python
model
```
Python

	Model	Score
0	Linear Regression	0.973684
1	Decision Tree Classifier	0.929825
2	RandomForest Classifier	0.964912
3	Gaussian Navie Bayes	0.964912
4	Multinomial Navie Bayes	0.850877