**Medicio – Blood Pressure Monitor**

**1. Abstract**

Medicio is a Flask-based web application for blood pressure monitoring and hypertension risk assessment. Users can input systolic, diastolic, and pulse rate values into a web interface, where the system predicts the blood pressure category using a trained machine learning model. It also provides educational resources for hypertension awareness. This project demonstrates the integration of machine learning, web development, and healthcare knowledge to produce a functional, user-friendly, and scalable health monitoring tool.

**2. Introduction**

**2.1 Project Overview**

Medicio aims to address the growing demand for accessible, reliable, and easy-to-use blood pressure monitoring tools. This web-based system enables real-time data entry, predictive analysis, and educational support for users. It is implemented using Python, Flask, HTML/CSS, Bootstrap, and a scikit-learn model.

**2.2 Purpose**

* Provide a convenient platform for entering and monitoring BP readings.
* Offer machine learning-based predictions for hypertension categories.
* Deliver educational content to improve cardiovascular health awareness.
* Serve as a foundation for future expansion into other healthcare metrics.

**3. Problem Statement**

Hypertension is a major contributor to cardiovascular diseases, yet many people lack tools for easy monitoring and predictive analysis. Commercial solutions can be costly, limited in functionality, or require special hardware.

**From different perspectives:**

* **End-user (Patient):** Needs quick, easy BP tracking with understandable results.
* **Healthcare Worker:** Requires reliable patient data for faster decision-making.
* **Developer:** Aims to design a scalable, accurate, and efficient monitoring system.

**4. Requirement Analysis**

**4.1 Functional Requirements**

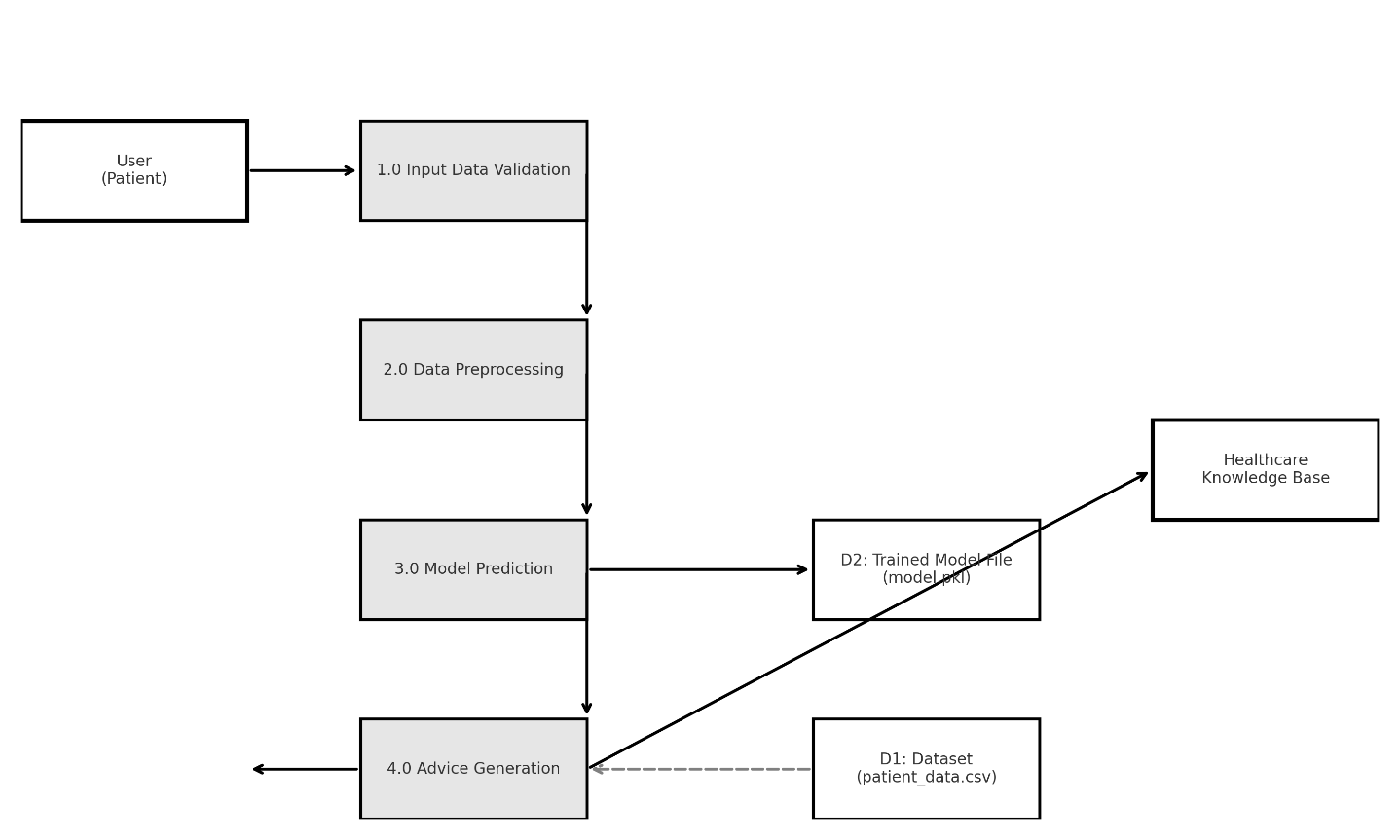
| **Functional Requirement** | **Description** |
| --- | --- |
| Data Input | User enters systolic, diastolic, and pulse rate values. |
| Prediction | System predicts BP category using ML model. |
| Data Storage | Stores readings for reference. |
| Education | Provides BP management resources. |
| Result Display | Clear display of predictions in UI. |

4.2 Non-Functional Requirements

| **Non-Functional Requirement** | **Description** |
| --- | --- |
| Usability | Intuitive interface with minimal steps. |
| Performance | Predictions generated in under 2 seconds. |
| Security | Safe handling of user input data. |
| Availability | Works on multiple devices and browsers. |
| Scalability | Easily extendable to include more features. |

**5. System Design**

The following diagrams illustrate the overall system structure and the data processing flow in Medicio.



**6. Technology Stack**

* **Frontend:** HTML, CSS, JavaScript (Bootstrap)
* **Backend:** Python (Flask)
* **Machine Learning:** scikit-learn, saved model.pkl
* **Data Storage:** CSV file (patient\_data.csv)
* **Tools:** Jupyter Notebook, GitHub

**7. Implementation**

**7.1 Web Application Workflow**

Flask handles routes for rendering HTML templates and processing form submissions. When the user submits BP readings, the backend loads the saved ML model (model.pkl), runs the prediction, and sends the result back to be displayed in the web interface.

**7.2 Machine Learning Model**

The model is trained on patient\_data.csv with features: systolic, diastolic, and pulse rate. Data preprocessing includes normalization and handling missing values. The classifier predicts hypertension stages and is saved for integration into Flask.

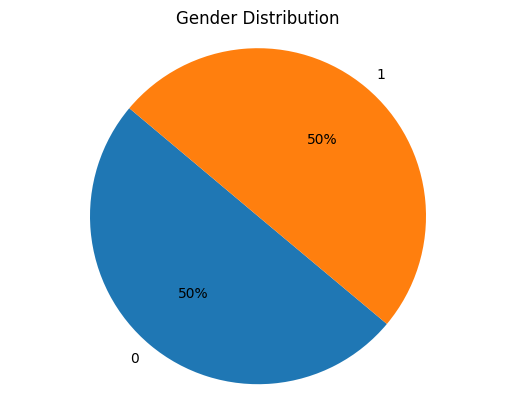
| **Stage** | **Feature** | **Description** | **Priority** | **Status** |
| --- | --- | --- | --- | --- |
| 1 | UI Development | Create HTML/CSS templates for input/output. | High | Completed |
| 2 | Model Training | Train ML model with dataset. | High | Completed |
| 3 | Backend Integration | Integrate Flask routes with model. | High | Completed |
| 4 | Testing | Validate model accuracy and UI performance. | Medium | Completed |
| 5 | Deployment | Host app locally or on server. | Medium | Pending |

9. Testing & Results

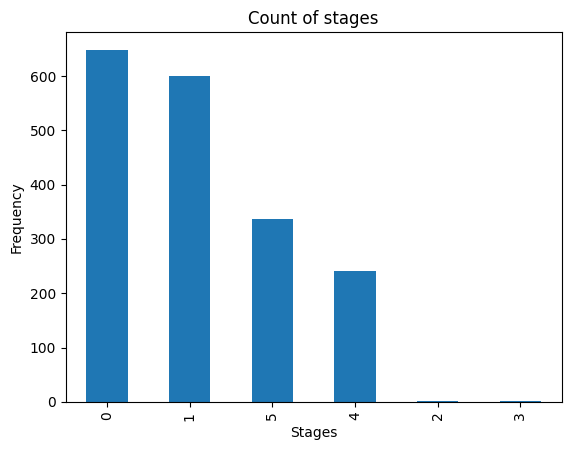
| **Test Case** | **Input** | **Expected Output** | **Actual Output** | **Status** |
| --- | --- | --- | --- | --- |
| TC1 | 120/80, Pulse 72 | Normal BP | Normal BP | Pass |
| TC2 | 140/95, Pulse 85 | Hypertension Stage 2 | Hypertension Stage 2 | Pass |
| TC3 | 130/85, Pulse 78 | Elevated BP | Elevated BP | Pass |

**In addition to the test case validations, exploratory data analysis and model performance metrics were visualized to better understand the dataset characteristics and model behavior.**

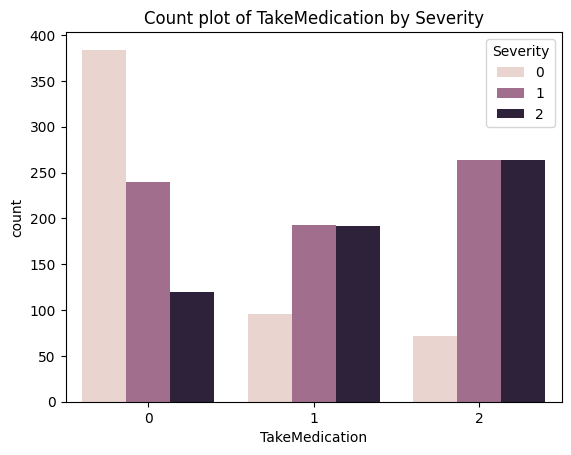
***Observation: Shows male/female ratio in the dataset.***



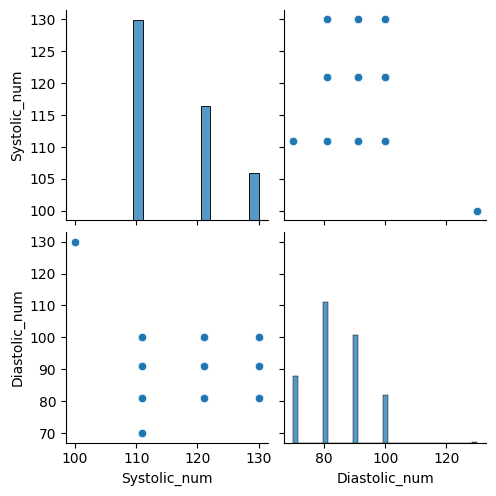
**Observation: Displays prevalence of each BP stage.**



**Observation: Shows relationship between medication usage and BP severity.**



**Observation: Highlights correlation patterns between the Systolic Readings and Diastolic Readings.**



**10. Advantages & Limitations**

**Advantages:**

* Accessible via any web browser.
* Quick predictions using ML.
* Educational content included.
* Easily extendable.

**Limitations:**

* Requires internet/server.
* Dependent on dataset quality.
* Not medically certified.

**11. Conclusion**

Medicio integrates machine learning with a user-friendly web interface to offer BP monitoring and hypertension risk prediction. It meets its goals of accessibility, usability, and scalability.

**12. Future Scope**

The current Medicio system provides a functional base for blood pressure monitoring and hypertension prediction. Future improvements can significantly expand its capabilities and impact:

* **IoT Device Integration** –  
  Connect Medicio with Bluetooth-enabled or Wi-Fi-enabled digital BP monitors to automatically capture readings without manual entry.
* **Mobile Application Development** –  
  Create Android and iOS apps to improve accessibility, allowing users to track readings and receive alerts on the go.
* **Cloud Storage & Sync** –  
  Store user data securely in the cloud so it can be accessed from multiple devices and backed up automatically.
* **Advanced Health Analytics** –  
  Extend the system to track additional metrics such as BMI, blood sugar levels, and cholesterol, providing a more holistic health profile.
* **Multi-Language Support** –  
  Implement regional language options to make the system more accessible to diverse user groups.
* **Predictive Health Alerts** –  
  Use continuous data trends and AI algorithms to alert users of potential health risks before symptoms appear.
* **Integration with Telemedicine Platforms** –  
  Allow healthcare professionals to remotely monitor patients’ BP data through secure dashboards, improving care in remote areas.

**13. References**

1. Medicio GitHub Repository: <https://github.com/Puru45-byte/Medicio---Blood-Pressure-Monitor>
2. Scikit-learn Documentation
3. Flask Documentation

**14. Appendix**

* GitHub Link: <https://github.com/Puru45-byte/Medicio---Blood-Pressure-Monitor>
* Dataset: patient\_data.csv
* Demo Video: demo.mp4