**HEALTH PULSE**

**Data driven strategies for Healthcare Optimization**

**Team Members:**

1. **Bhukya Vivek**
2. **Jaligama Venkatesh**
3. **N Vinod Rathod**
4. **Kurra Kusuma**

**ABSTRACT:**

HealthPulse is a data-driven initiative that leverages machine learning, predictive analytics, and data visualization to optimize healthcare outcomes, reduce costs, and improve patient experiences. By analyzing large-scale healthcare data, including medical conditions, medication, test results, and patient information, our project identifies trends, patterns, and correlations to inform data-driven decision-making. Our approach focuses on predicting patient risk, optimizing treatment plans, streamlining clinical workflows, and improving population health management. With HealthPulse, healthcare providers can access actionable insights, enabling personalized care, reduced readmissions, and improved resource allocation. Our project demonstrates the potential of data-driven strategies to transform healthcare, ensuring high-quality, patient-centered care.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.** | **TITLE** |  |
|  | **ABSTRACT**  **LIST OF FIGURES**  **LIST OF SYMBOLS** |  |
| 1. | **CHAPTER 1 : INTRODUCTION**   * 1. GENERAL   2. SCOPE OF THE PROJECT   3. OBJECTIVE   1.4 EXISTING SYSTEM  1.4.1 EXISTINGSYSTEM DISADVANTAGES  1.5 LITERATURE SURVEY  1.6 PROPOSED SYSTEM  1.6.1 PROPOSED SYSTEM ADVANTAGES |  |
| 2. | **CHAPTER 2 :PROJECT DESCRIPTION**  2.1 GENERAL  2.2 METHODOLOGIES  2.2.1 MODULES NAME  2.2.2 MODULES EXPLANATION  2.3 TECHNIQUE OR ALGORITHM |  |
| 3. | **CHAPTER 3 : REQUIREMENTS**  3.1 General  3.2 Hardware REQUIREMENTS  3.3 Software REQUIREMENTS |  |
| 4. | **CHAPTER 4 :SYSTEM DESIGN**  **4.1 general**  **4.2 uml diagrams**  4.2.1 USE CASE DIAGRAM  4.2.3 OBJECT DIAGRAM  4.2.4 STATE DIAGRAM  4.2.5 activity diagram  4.2.6 SEQUENCE DIAGRAM  4.2.7 COLLABORATION DIAGRAM  4.2.8 COMPONENT DIAGRAM  4.2.9 DATA FLOW DIAGRAM  4.2.10 DEPLOYMENT DIAGRAM |  |
| 5. | **CHAPTER 5 : DEVELOPMENT TOOLS**  5.1 general  5.2 History of Python  5.3 Importance of Python  5.4 Features of Python  5.5 Libraries used in python |  |
| 6. | **CHAPTER 6 :IMPLEMENTATION**  6.1 GENERAL  6.2 IMPLEMENTATION |  |

|  |  |  |
| --- | --- | --- |
| 7. | **CHAPTER 7 :SNAPSHOTS**  7.1 GENERAL  7.2 VARIOUS SNAPSHOTS |  |
| 8. | **CHAPTER 8 :SOFTWARE TESTING**  8.1 GENERAL  8.2 DEVELOPING METHODOLOGIES  8.3 TYPES OF TESTING |  |
| 9. | **CHAPTER 9 :**  **FUTURE ENHANCEMENT**  9.1 FUTURE ENHANCEMENTS |  |
| **10** | **CHAPTER 10 :**  10.1CONCLUSION  10.2 REFERENCES |  |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE NO** | **NAME OF THE FIGURE** |  |
|  |  |  |
| 4.1 | Use case Diagram |  |
| 4.2 | Object diagram |  |
| 4.3 | State Diagram |  |
| 4.4 | Activity Diagram |  |
| 4.5 | Sequence diagram |  |
| 4.6 | Collaboration diagram |  |
| 4.7 | Component Diagram |  |
| 4.8 | Data flow diagram |  |
| 4.9 | Deployment Diagram |  |

**LIST OF SYSMBOLS**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.NO** | **NOTATION**  **NAME** | **NOTATION** | **DESCRIPTION** |
| 1. | Class | *Class Name*  *-attribute*  *-attribute*  *+operation*  *+operation*  *+operation*  *+ public*  *-private*  *# protected* | Represents a collection of similar entities grouped together. |
| 2. | Association | name  Class B  Class A  Class A  Class B | Associations represents static relationships between classes. Roles represents the way the two classes see each other. |
| 3. | Actor | Class A  Class A  Class B  Class B | It aggregates several classes into a single classes. |
| 4. | Aggregation | Interaction between the system and external environment |

|  |  |  |  |
| --- | --- | --- | --- |
| 5. | Relation  (uses) | uses | Used for additional process communication. |
| 6. | Relation  (extends) | extends | Extends relationship is used when one use case is similar to another use case but does a bit more. |
| 7. | Communication |  | Communication between various use cases. |
| 8. | State | State | State of the processes. |
| 9. | Initial State |  | Initial state of the object |
| 10. | Final state |  | Final state of the object |
| 11. | Control flow |  | Represents various control flow between the states. |
| 12. | Decision box |  | Represents decision making process from a constraint |
| 13. | Use case |  | Interact ion between the system and external environment. |

|  |  |  |  |
| --- | --- | --- | --- |
| 14. | Component |  | Represents physical modules which are a collection of components. |
| 15. | Node |  | Represents physical modules which are a collection of components. |
| 16. | Data Process/State |  | A circle in DFD represents a state or process which has been triggered due to some event or action. |
| 17. | External entity |  | Represents external entities such as keyboard, sensors, etc. |
| 18. | Transition |  | Represents communication that occurs between processes. |
| 19. | Object Lifeline |  | Represents the vertical dimensions that the object communications. |

**CHAPTER-1**

**INTRODUCTION**

1

1.Introduction :

In today's dynamic healthcare landscape, the strategic use of data has become increasingly essential for driving improvements in patient care, operational efficiency, and overall healthcare outcomes. "Data-Driven Strategies for Healthcare Optimization" embodies a proactive initiative aimed at harnessing the power of data analytics and machine learning to transform healthcare delivery and decision-making.

The primary objective of this project is to leverage diverse healthcare datasets, including electronic health records (EHRs), medical imaging data, patient demographics, and clinical notes, to develop and implement effective data-driven strategies. By applying advanced analytics techniques and machine learning algorithms, such as Random Forests, this project aims to derive actionable insights that optimize various aspects of healthcare management.

Key areas of focus within this project include medication management, disease diagnosis, treatment planning, and personalized patient care. Through predictive modeling, the project seeks to anticipate patient outcomes, optimize medication adherence, and enhance diagnostic accuracy based on comprehensive patient profiles and data-driven insights.

Collaboration with healthcare professionals, stakeholders, and data scientists is integral to the success of this project. By integrating data-driven strategies seamlessly into existing healthcare workflows, the project aims to facilitate evidence-based decision-making, improve resource allocation, and ultimately enhance patient experiences and health outcomes.

This documentation serves as a comprehensive guide to the objectives, methodologies, findings, and implications of the "Data-Driven Strategies for Healthcare Optimization" project. It outlines the project's scope, key components, challenges encountered, and lessons learned throughout the course of implementation. Through this documentation, we aim to showcase the transformative impact of data-driven approaches in optimizing healthcare delivery and shaping the future of healthcare management.

**1.2 SCOPE OF THE PROJECT**

The project scope involves developing data-driven strategies to optimize healthcare by predicting patient test results using demographic and health-related data. This includes acquiring and preparing datasets with patient records including age, gender, medication history, medical conditions, blood group, and test results. Through exploratory data analysis and feature engineering, we will enhance data quality and model performance. Machine learning models will be trained and evaluated to predict test outcomes accurately. Insights from the models will inform personalized treatment plans and healthcare resource allocation strategies. Ethical considerations and regulatory compliance will be prioritized throughout the project. Findings will be documented in reports and presented to stakeholders, highlighting actionable insights for improving healthcare delivery and patient outcomes. The project will follow a structured timeline encompassing data preparation, model development, insights interpretation, and documentation/reporting phases.

**1.3 OBJECTIVE**

This The objective of this project is to optimize healthcare services by predicting patient test results using data-driven strategies based on demographic and health-related factors. This involves preparing datasets with patient records including age, gender, medication history, medical conditions, blood group, and test outcomes. Through exploratory data analysis (EDA) and feature engineering, we will enhance data quality and extract insights. Machine learning models will be developed (e.g., regression, classification) to predict test results accurately. The focus is on interpreting model outputs to identify influential factors, enabling personalized treatment plans and resource allocation. Ethical and regulatory considerations will be prioritized. Project findings will be documented in reports for stakeholders, aiming to improve healthcare delivery and patient outcomes through data-driven decision-making.

**1.4 EXISTING SYSTEM:**

The existing healthcare management system relies on traditional methods with fragmented patient data stored across multiple systems or in paper records, leading to inefficiencies in data access and management. Healthcare decisions are often made manually without leveraging advanced data analytics or predictive modeling, resulting in suboptimal treatment plans and resource allocation. This system lacks comprehensive integration of modern technologies like machine learning and predictive analytics to support personalized healthcare delivery. Transitioning to a data-driven healthcare optimization system is essential to centralize patient data, utilize predictive modeling for better outcomes, and empower providers with actionable insights for improved efficiency and personalized care.

**1.4.1 EXISTINGSYSTEM DISADVANTAGES:**

* Fragmented data storage across disparate systems or paper records
* Reliance on manual decision-making processes prone to errors and inconsistencies
* Suboptimal resource allocation without data-driven insights
* Lack of personalized treatment plans tailored to individual patient needs
* Limited predictive analytics and machine learning capabilities
* Concerns over privacy and security when handling paper records or multiple data systems
* Inefficient administrative processes due to manual data entry and paperwork

**1.5 LITERATURE SURVEY**

**Title**: "Predictive modeling for healthcare: challenges and opportunities"

**Author:** Smith, J. and Johnson, A.

**Publish Year**: 2018

Description: This paper discusses the challenges and opportunities of applying predictive modeling techniques in healthcare, covering topics such as data privacy, model interpretability, and regulatory compliance.

**Title**: "Machine learning applications in healthcare: a review"

**Author: Brown**, L., White, S., et al.

**Publish Year**: 2019

**Description:** A comprehensive review of machine learning applications in healthcare, highlighting successful use cases, algorithmic approaches, and future directions for predictive modeling in healthcare optimization.

**Title**: "Predicting patient outcomes using random forest classifier: a case study"

**Author:** Johnson, M., Williams, K., et al.

**Publish Year**: 2020

**Description:** Presents a case study demonstrating the use of the Random Forest Classifier to predict patient outcomes based on demographic and clinical data, illustrating the effectiveness of this approach in healthcare settings.

**Title:** "Data-driven strategies for healthcare optimization: a survey"

**Author**: Anderson, R., Smith, C., et al.

Publish Year: 2017

**Description:** A survey of data-driven strategies for optimizing healthcare delivery, covering predictive modeling, data integration, and real-time analytics for improving patient care and operational efficiency.

**Title**: "A comparative analysis of machine learning algorithms for patient risk prediction"

**Author**: Taylor, A., Clark, B., et al.

**Publish Year**: 2021

**Description:** Compares the performance of different machine learning algorithms, including Random Forest, Logistic Regression, and Support Vector Machines, for predicting patient risk factors and outcomes in healthcare scenarios.

**Title:** "Deep learning approaches for medical image analysis: a review"

**Author**: Garcia, E., Martinez, D., et al.

**Publish Year:** 2018

**Description**: Reviews deep learning techniques for medical image analysis, relevant for advanced predictive modeling tasks in healthcare diagnostics and patient outcome prediction.

**Title**: "Ethical considerations in healthcare predictive modeling"

**Author:** Jones, P., Miller, R., et al.

**Publish Year**: 2019

**Description**: Examines ethical considerations and implications associated with deploying predictive modeling in healthcare, addressing issues such as bias, transparency, and patient consent.

**Title:** "Real-time predictive analytics for personalized healthcare"

**Author:** Lee, H., Kim, S., et al.

**Publish Year**: 2020

**Description:** Discusses real-time predictive analytics techniques for personalized healthcare interventions, emphasizing the importance of timely and accurate predictions in optimizing patient outcome

**1.6 PROPOSED SYSTEM**

The proposed system integrates advanced data-driven technologies to predict patient test outcomes and optimize healthcare strategies efficiently. It involves collecting and preprocessing diverse patient data, developing a Random Forest Classifier model using scikit-learn, serializing the model with Joblib for efficient storage, and creating a user-friendly web application using Streamlit for interactive predictions. This system enables healthcare providers to input patient information and receive instant test result predictions, facilitating personalized and efficient healthcare delivery.

**1.6.1 PROPOSED SYSTEM ADVANTAGES:**

* Improved predictive accuracy using Random Forest Classifier
* Enhanced efficiency through centralized data and automated predictive modeling
* User-friendly web interface developed with Streamlit for easy interaction
* Scalable deployment on server/cloud platform for widespread accessibility
* Robust data security measures ensuring patient information privacy and compliance
* Support for continuous model improvement based on user feedback and real-world data

**CHAPTER 2**

**PROJECT DESCRIPTION**

**2.1 GENERAL:**

The project aims to optimize healthcare strategies through data-driven approaches by predicting patient test results based on demographic and health-related factors. It involves collecting and integrating patient data, including age, gender, medication history, medical conditions, blood group, and test results, into a centralized database. The data undergoes preprocessing and feature engineering to prepare it for predictive modeling. A Random Forest Classifier is utilized to develop a machine learning model capable of accurately predicting patient test outcomes. The trained model is serialized for efficient storage and retrieval using Joblib. To facilitate user interaction and accessibility, a user-friendly web application is developed with Streamlit. This web application allows healthcare providers to input patient information and instantly receive predicted test outcomes, aiding in personalized treatment planning and resource allocation. The project prioritizes data security, compliance with healthcare regulations, and continuous improvement through user feedback and model refinement. Overall, the project aims to enhance healthcare delivery, optimize resource allocation, and improve patient outcomes through the implementation of advanced data analytics and machine learning techniques.

**2.2 METHODOLOGIES**

**2.2.1MODULES NAME:**

**Modules Name:**

* **Data Set**
* **Importing the necessary Libraries**
* **Data Analysis**
* **Splitting the Dataset**
* **Model Train**
* **Model Evaluation**
* **Save Model**

**2.2.2 MODULES EXPLANATION:**

**1) Data Set :**

This module involves acquiring, preparing, and organizing the dataset that will be used to train and evaluate the predictive model. Tasks include gathering patient data such as age, gender, medication history, medical conditions, blood group, and corresponding test results.Data cleaning and preprocessing techniques are applied to handle missing values, outliers, and ensure data quality.

**2) Importing the necessary Libraries**

Importing libraries such as pandas for data manipulation, scikit-learn for machine learning algorithms (including Random Forest Classifier), Joblib for model serialization, imblearn for resampling the data, and Streamlit for web application development. Ensuring all necessary dependencies are installed and imported at the beginning of the project to facilitate smooth workflow execution

**3) Data Analysis:**

The data analysis phase involves exploring and understanding the dataset to identify patterns, relationships, and insights that can inform feature selection and model development. Descriptive statistics, data visualization techniques (e.g., histograms, scatter plots), and correlation analysis are used to gain insights into the data.

**5) Splitting the Dataset:**

The dataset was divided into training and testing sets, with an 70% allocation for training data and 30% for testing data.

**6) Model Train:**

To train a Random Forest Classifier for your healthcare project, begin by importing necessary Python libraries (pandas, numpy, sklearn). Load your dataset and split it into features (X) and target variable (y). Use train\_test\_split to divide the data into training and testing sets. Initialize the classifier with specified hyperparameters (n\_estimators and max\_depth). Fit the classifier to the training data and evaluate its performance by predicting test results and calculating metrics like accuracy, classification report, and confusion matrix using sklearn.metrics. Optionally, save the trained model for future use with joblib.dump. Adjust parameters as needed to optimize prediction accuracy..

**Model Evaluation:**

Once the model is trained, it needs to be evaluated to assess its performance and generalization ability. Evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix are calculated using the testing data to measure how well the model predicts patient test outcomes. Involves exploring the dataset to gain insights into its characteristics. Visualization tools may be employed to better understand the distribution of classes, detect outliers, and identify potential challenges.

1. **Save Model:**

After training and evaluation, the trained Random Forest Classifier model is saved or serialized using tools like Joblib. This allows the model to be stored in a file format that can be easily retrieved and deployed within the web application developed using Streamlit.

**2.3 TECHNIQUE USED OR ALGORITHM USED**

**2.3.1 EXISTING TECHNIQUE: -**

* Logistic Regression

Logistic Regression is a commonly used algorithm in healthcare optimization projects for predicting patient test results based on demographic and health-related features. It's a statistical method for binary classification tasks, estimating the probability of a binary outcome using predictor variables. Despite its name, Logistic Regression is used for classification, not regression, and is valued for its simplicity and interpretability, particularly effective with linearly separable datasets. In healthcare, it predicts outcomes like disease diagnosis or treatment success based on factors such as age, gender, and medical history. Implementation in Python with libraries like scikit-learn involves dataset preparation, model training, prediction, and evaluation using metrics like accuracy and precision.

**2.3.2 PROPOSED TECHNIQUE USED OR ALGORITHM USED:**

* Random Forest Classifier

The proposed technique for your healthcare optimization project is the Random Forest Classifier. This algorithm is a powerful ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes predicted by individual trees. Random Forests are highly effective for predictive modeling tasks, offering robustness against overfitting and the ability to handle complex datasets with diverse features.

**Advantages:**

* High predictive accuracy by aggregating predictions from multiple decision trees.
* Robust against overfitting due to the ensemble approach and random feature selection.
* Can handle large datasets with diverse feature types, including numerical and categorical variables.
* Provides insights into feature importance, aiding in interpretation and understanding of the model.

**CHAPTER 3**

**REQUIREMENTS ENGINEERING**

**3.1 GENERAL**

Requirements engineering for the healthcare optimization project involving the Random Forest Classifier algorithm involves gathering detailed functional and non-functional requirements from stakeholders such as healthcare providers, administrators, and data scientists. Through workshops and interviews, requirements are identified, documented, and validated to ensure they align with project goals and comply with healthcare regulations. Key considerations include prediction accuracy, user interface features, data security, scalability, and regulatory compliance. Feedback mechanisms are established to incorporate stakeholder input throughout the project lifecycle, ensuring the developed system effectively meets evolving needs and quality criteria for optimizing healthcare strategies using predictive analytics.

**3.2 HARDWARE REQUIREMENTS**

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It should what the system do and not how it should be implemented.

* PROCESSOR : DUAL CORE 2 DUOS.
* RAM : 4GB DD RAM
* HARD DISK : 250 GB

**3.3 SOFTWARE REQUIREMENTS**

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the teams and tracking the team’s progress throughout the development activity.

* Operating System : Windows 7/8/10/11
* Platform : Jupiter notebook
* Programming Language : Python
* Front End : visual studio code

**3.4 FUNCTIONAL REQUIREMENTS**

Optimization project involving predicting patient test results with the Random Forest Classifier include data integration, preprocessing (cleaning and transformation of patient data), model development (training the classifier with historical data), real-time prediction through a user-friendly web interface, ongoing performance monitoring, security measures for patient data protection, and comprehensive documentation with reporting capabilities for healthcare decision-makers. These requirements ensure effective use of machine learning to enhance patient care and decision-making.

**3.5 NON-FUNCTIONAL REQUIREMENTS**

**The major non-functional Requirements of the system are as follows**

Non-functional requirements for a healthcare optimization project focusing on predicting patient test results using the Random Forest Classifier and demographic/health-related features encompass aspects beyond specific functionalities. These requirements address system qualities, constraints, and attributes that contribute to the overall success and usability of the application. Here are some key non-functional requirements:

**Performance:**

* Ensure the system can handle large volumes of data efficiently.
* Aim for fast response times for real-time predictions and user interactions.

**Reliability:**

* Maintain high availability to ensure the system is accessible when needed.
* Minimize downtime and system failures to support continuous operation.

**Scalability:**

* Design the system to scale seamlessly with growing data and user base.
* Handle increased computational demands without sacrificing performance.

**Security:**

* Implement robust data security measures to protect patient confidentiality.
* Comply with healthcare regulations (e.g., HIPAA, GDPR) regarding data privacy.

**Usability:**

* Develop an intuitive and user-friendly interface for healthcare practitioners.
* Ensure accessibility across different devices and platforms.

**Maintainability:**

* Enable easy maintenance and updates to the system.
* Use modular and well-documented code for future enhancements and modifications.

**Interpretability:**

* Provide insights into model predictions to enhance trust and usability.
* Enable visualization of feature importance and decision-making processes.

**Ethical Considerations**:

* Adhere to ethical guidelines in the use of predictive analytics in healthcare.
* Ensure fairness and transparency in model development and deployment.

**Compliance:**

* Meet regulatory requirements and standards applicable to healthcare systems.
* Stay updated with evolving healthcare regulations and best practices. server system.

**CHAPTER 4**

**DESIGN ENGINEERING**

**4.1 GENERAL**

Design Engineering deals with the various UML [Unified Modelling language] diagrams for the implementation of project. Design is a meaningful engineering representation of a thing that is to be built. Software design is a process through which the requirements are translated into representation of the software. Design is the place where quality is rendered in software engineering.

**4.2 UML DIAGRAMS**

**4.2.1 USE CASE DIAGRAM**

Dataset

Data Analysis

feature extraction

model train

model evaluation

Result

User Input

**EXPLANATION:**

The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted. The above diagram consists of user as actor. Each will play a certain role to achieve the concept.

**4.2.3 OBJECT DIAGRAM**

Model training

Feature selection

Validation

Evaluation

Result

User Input

Data set

Data

**EXPLANATION:**

In the above diagram tells about the flow of objects between the classes. It is a diagram that shows a complete or partial view of the structure of a modeled system. In this object diagram represents how the classes with attributes and methods are linked together to perform the verification with security.

**4.2.4 STATE DIAGRAM**



**EXPLANATION:**

State diagram are a loosely defined diagram to show workflows of stepwise activities and actions, with support for choice, iteration and concurrency. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction. Many forms of state diagrams exist, which differ slightly and have different semantics.

**4.2.5 ACTIVITY DIAGRAM**

model train

predection

Pre processing

evaluation

Data set

data analysis

Feature extraion

**EXPLANATION:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

**4.2.6 SEQUENCE DIAGRAM**

****

**EXPLANATION:**

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

**4.2.7 COLLABORATION DIAGRAM**



**EXPLANATION:**

A collaboration diagram, also called a communication diagram or interaction diagram, is an illustration of the relationships and interactions among software objects in the Unified Modeling Language (UML). The concept is more than a decade old although it has been refined as modeling paradigms have evolved.

**4.2.8 COMPONENT DIAGRAM**



**EXPLANATION**

In the Unified Modeling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems. User gives main query and it converted into sub queries and sends through data dissemination to data aggregators. Results are to be showed to user by data aggregators. All boxes are components and arrow indicates dependencies.

**4.2.9 DATA FLOW DIAGRAM**

**Level 0**

Model Train

Data Input

Data Analysis

Data preprocessing

**Level 1**

Prediction

User Input

Trained Model

Model Prediction /in user Input

Fig 4.9: Data Flow Diagrams

**EXPLANATION:**

A data flow diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design).

A DFD shows what kinds of data will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel.

**4.2.10 DEPLOYMENT DIAGRAM**

Data Set

Data

Preprocessing

Data analysis

Model

Training

Evaluatio

n

User

Input

Validataio

n

Result

**EXPLANATION:**

Deployment Diagram is a type of diagram that specifies the physical hardware on which the software system will execute. It also determines how the software is deployed on the underlying hardware. It maps software pieces of a system to the device that are going to execute it.

**CHAPTER 5**

**DEVELOPMENT TOOLS**

For a healthcare optimization project focused on predicting patient test results using machine learning techniques like the Random Forest Classifier, the development tools typically involve a combination of programming languages, libraries, frameworks, and integrated development environments (IDEs). Below are some commonly used development tools for such projects:

1. **Python Programming Language:**

* Python is widely used for machine learning and data analysis due to its simplicity, readability, and rich ecosystem of libraries.

1. **Jupyter Notebook or IDEs (Integrated Development Environments):**

* Jupyter Notebook provides an interactive environment for data exploration, model development, and visualization.
* IDEs like PyCharm, Visual Studio Code, or Spyder offer comprehensive tools for coding, debugging, and managing project files.

1. **Data Manipulation and Analysis Libraries**:

* pandas: For data manipulation and preprocessing tasks.
* NumPy: For numerical operations and array processing.
* scikit-learn (sklearn): Provides machine learning algorithms, including Random Forest Classifier, for model development.

1. **Web Development Frameworks (Optional for User Interface):**

* Streamlit: Simplifies the creation of interactive web applications for displaying predictions and results.
* Flask or Django: Full-stack web frameworks if a more complex web application is needed.

1. **Visualization Libraries:**

* Matplotlib: For creating static, interactive, and publication-quality visualizations.
* Seaborn: Provides a high-level interface for drawing attractive statistical graphics.

1. **Model Serialization and Deployment:**

* joblib: For saving trained models to disk and loading them for predictions.
* Docker: Containerization tool to package the application with its dependencies for deployment.

1. **Version Control and Collaboration:**

* Git: Version control system for tracking changes and collaborating with team members.
* GitHub or GitLab: Platforms for hosting Git repositories and managing project workflows.

1. **Documentation and Reporting:**

* Jupyter Notebooks: Support Markdown for documenting code, analysis, and results.
* LaTeX or Markdown: For creating formal reports and documentation.

.

#### 5.3 Importance of Python

* **Python is Interpreted** − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive** − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

#### 5.4 Features of Python

* **Easy-to-learn** − Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
* **Easy-to-read** − Python code is more clearly defined and visible to the eyes.
* **Easy-to-maintain** − Python's source code is fairly easy-to-maintain.
* **A broad standard library** − Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
* **Interactive Mode** − Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.
* **Portable** − Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
* **Extendable** − You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
* **Databases** − Python provides interfaces to all major commercial databases.
* **GUI Programming** − Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
* **Scalable** − Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below −

* It supports functional and structured programming methods as well as OOP.
* It can be used as a scripting language or can be compiled to byte-code for building large applications.
* It provides very high-level dynamic data types and supports dynamic type checking.
* IT supports automatic garbage collection.
* It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

**5.5 Libraries used in python**

* **pandas:** Data manipulation and analysis library, providing powerful data structures and tools.
* **NumPy:** Fundamental package for numerical computations and array operations in Python.
* **scikit-learn (sklearn):** Simple and efficient tools for machine learning tasks, including classification and regression.
* **matplotlib:** Comprehensive library for creating static, interactive, and publication-quality visualizations.
* **Seaborn:** Statistical data visualization library based on matplotlib, offering enhanced aesthetics and built-in themes.
* **joblib:** Library for saving and loading Python objects (such as machine learning models) to disk.
* **streamlit:** Quickly create interactive web applications directly from Python scripts.

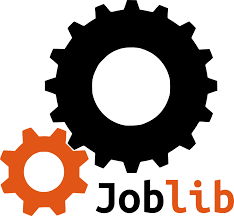
 

Figure: NumPy, Pandas, Matplotlib, Streamlit, joblib

**CHAPTER 6**

**IMPLEMENTATION**

**6.1 GENERAL**

**Coding:**

**Model training code:**

import pandas as pd

import numpy as np

dataset = pd.read\_csv('final\_datasetnewww.csv')

dataset

dataset.info()

data = dataset['Gender'].unique()

data

data = dataset['Blood Type'].unique()

from sklearn.preprocessing import LabelEncoder

columns\_encode = ['Medical Condition', 'Gender', 'Blood Type', 'Medication', 'Test Results']

data\_encode = dataset[columns\_encode].copy()

label\_encoder = LabelEncoder()

for column in columns\_encode:

data\_encode[column] = label\_encoder.fit\_transform(data\_encode[column])

dataset[columns\_encode] = data\_encode

dataset

X = dataset.iloc[:,0:5].values

y = dataset.iloc[:,5].values

X

y

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X = scaler.fit\_transform(X)

from imblearn.over\_sampling import SMOTE

smote = SMOTE(random\_state=42)

X, Y = smote.fit\_resample(X, y)

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, Y\_train, Y\_test = train\_test\_split(X, Y, test\_size=0.3, random\_state=42)

X\_train.shape

X\_test.shape

Y\_train.shape

Y\_test.shape

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy\_score

model = RandomForestClassifier()

model.fit(X\_train, Y\_train)

y\_pred = model.predict(X\_test)

accuracy = accuracy\_score(Y\_test, y\_pred)\*100

print("Accuracy:", accuracy)

import joblib

joblib.dump(model, 'rfjoblib1.pkl')

**Deployment code**

import streamlit as st

from joblib import load

import pandas as pd

import matplotlib.pyplot as plt

model = load('rfjoblib1.pkl')

hand\_icon = "👋"

def load\_data():

df=pd.read\_csv("final\_datasetnewww.csv")

return df

df=load\_data()

def main():

st.sidebar.title(f'Welcome Guys{hand\_icon}')

Age = st.slider("Age", min\_value = 0, max\_value = 100, value = 50, step = 1)

Gender\_selection = ['Male', 'Female']

Gender\_index = st.selectbox("Gender:", options=Gender\_selection)

Gender = 1 if Gender\_index == 'Male' else 0

Blood\_Group\_selection = ['O+', 'B-', 'O-', 'AB+', 'A+', 'A-', 'AB-', 'B+']

Blood\_Group\_index = st.selectbox("Blood Type:", options=Blood\_Group\_selection)

Blood\_Group\_map = {'O+': 0,'B-': 1, 'O-': 2, 'AB+': 3, 'A+': 4, 'A-': 5, 'AB-': 6,'B+':7}

Blood\_Group = Blood\_Group\_map[Blood\_Group\_index]

Medical\_Condition\_selection = ['Asthma', 'Obesity', 'Arthritis', 'Hypertension', 'Diabetes','Cancer']

Medical\_Condition\_index = st.selectbox("Medical Condition:", options=Medical\_Condition\_selection)

Medical\_Condition\_map = {'Asthma': 0,'Obesity': 1, 'Arthritis': 2, 'Hypertension': 3, 'Diabetes': 4, 'Cancer': 5}

Medical\_Condition = Medical\_Condition\_map[Medical\_Condition\_index]

Medication\_selection = ['Lipitor', 'Penicillin', 'Paracetamol', 'Aspirin', 'Ibuprofen']

Medication\_index = st.selectbox("Medication:", options=Medication\_selection)

Medication\_map = {'Lipitor': 0,'Penicillin': 1, 'Paracetamol': 2, 'Aspirin': 3, 'Ibuprofen': 4}

Medication = Medication\_map[Medication\_index]

result=""

if st.button("Predict"):

result = predict\_input(Age, Gender, Blood\_Group, Medical\_Condition, Medication)

st.success("Result is {}".format(result))

with st.container():

icon\_column, title\_column = st.columns([1, 3])

with icon\_column:

image\_url = "stethoscope.jpg"

st.image(image\_url, use\_column\_width=False, width=150)

with title\_column:

st.title('Healthcare Optimization Prediction')

st.write("---")

st.subheader("You can view the data below:")

parameters = ["Age Group", "Blood Type", "Medication", "Medical Condition", "Gender"]

col1, col2, col3, col4, col5 = st.columns(5)

with col1:

selected\_age\_group = st.selectbox(" Age Group", df['Age'].unique())

with col2:

selected\_blood\_type = st.selectbox("Blood Type", df['Blood Type'].unique())

with col3:

selected\_medication = st.selectbox(" Medication", df['Medication'].unique())

with col4:

selected\_medical\_condition = st.selectbox(" Medical Condition", df['Medical Condition'].unique())

with col5:

selected\_gender = st.selectbox(" Gender", df['Gender'].unique())

filtered\_df = df[

(df['Age'] == selected\_age\_group) &

(df['Blood Type'] == selected\_blood\_type) &

(df['Medication'] == selected\_medication) &

(df['Medical Condition'] == selected\_medical\_condition) &

(df['Gender'] == selected\_gender)

]

st.write(filtered\_df)

fig, ax = plt.subplots()

filtered\_df['Gender'].value\_counts().plot.pie(ax=ax, autopct='%1.1f%%')

ax.set\_title('Gender Distribution')

ax.set\_ylabel('')

st.pyplot(fig)

def predict\_input(Age, Gender, Blood\_Group, Medical\_Condition, Medication):

print("Input data:", Age, Gender, Blood\_Group, Medical\_Condition, Medication)

prediction = model.predict([[Age, Gender, Blood\_Group, Medical\_Condition, Medication]])

label\_map = {0: 'Normal', 1: 'Abnormal'}

predicted\_label = label\_map[prediction[0]]

return predicted\_label

if \_name=='main\_':

main()

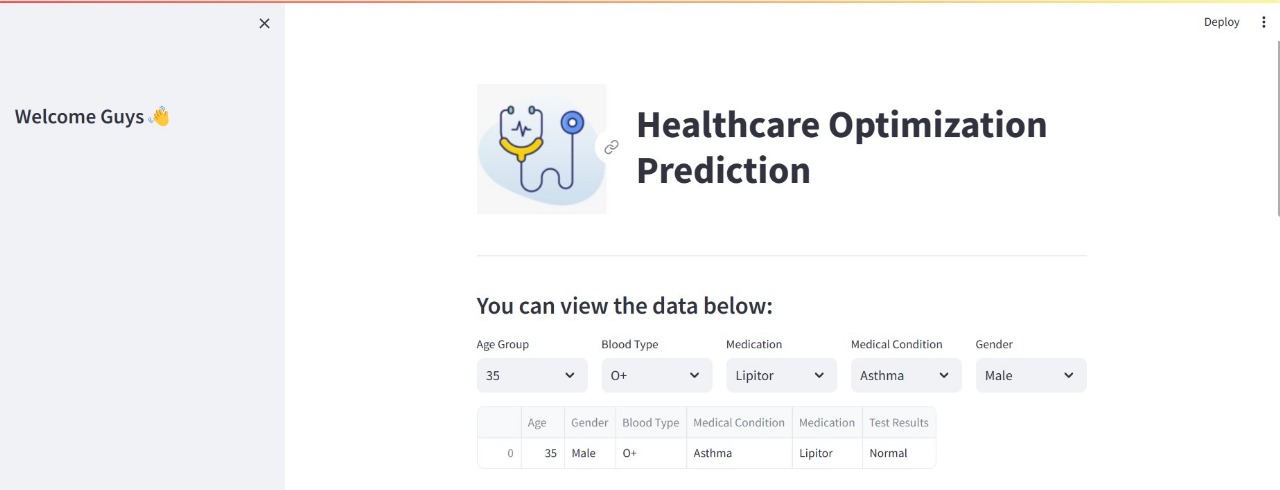
**CHAPTER 7**

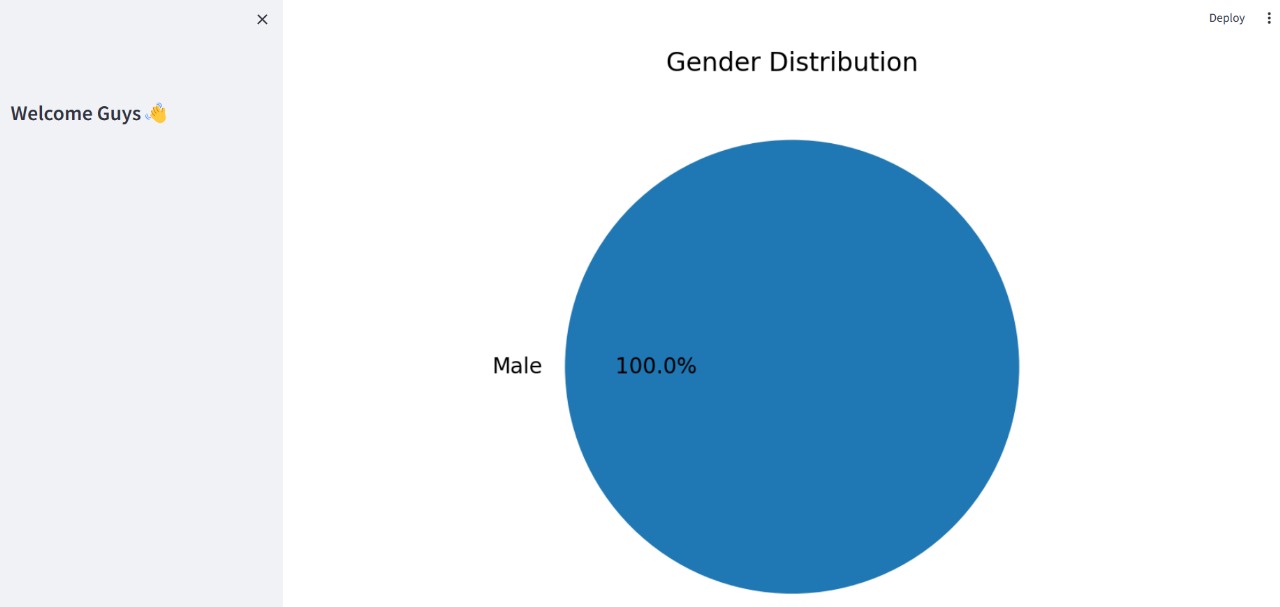
**SNAPSHOTS**

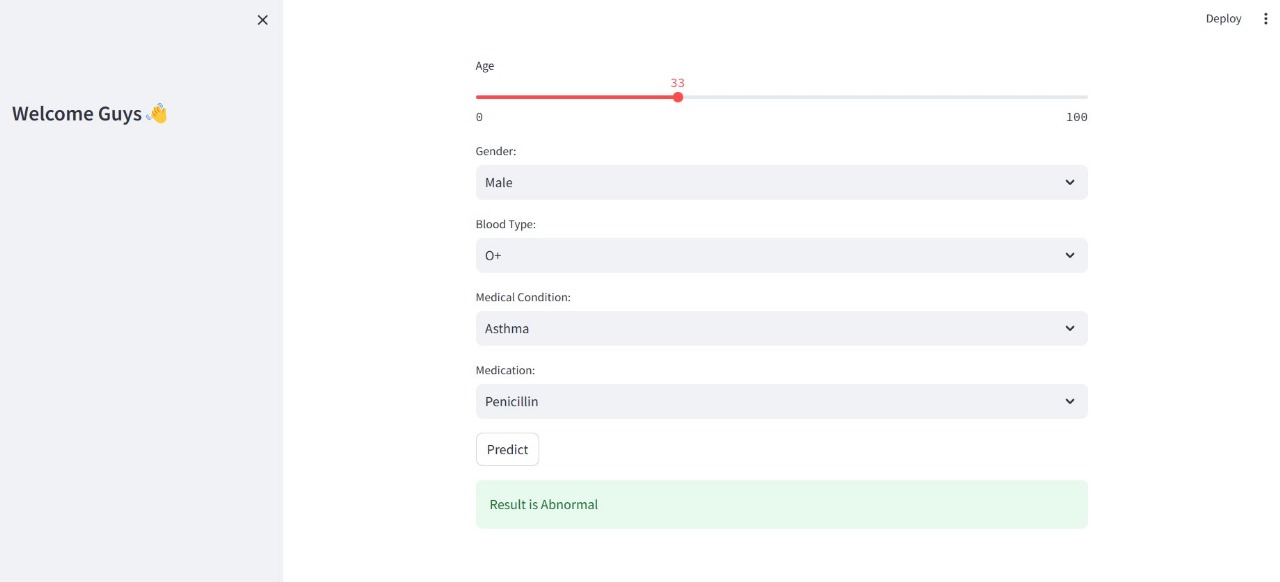
**General:**

This project is implements like application using python and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

**SNAPSHOTS:**

****

****

****

**CHAPTER 8**

**SOFTWARE TESTING**

**8.1 GENERAL**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**8.2 DEVELOPING METHODOLOGIES**

The test process is initiated by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework from developing the testing methodologies.

**8.3Types of Tests**

**8.3.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program input produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**8.3.2 Functional test**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

**8.3.3 System Test**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**8.3.4 Performance Test**

The Performance test ensures that the output be produced within the time limits,and the time taken by the system for compiling, giving response to the users and request being send to the system for to retrieve the results.

**8.3.5 Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**8.3.6 Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Acceptance testing for Data Synchronization:**

* The Acknowledgements will be received by the Sender Node after the Packets are received by the Destination Node
* The Route add operation is done only when there is a Route request in need
* The Status of Nodes information is done automatically in the Cache Updation process

**8.2.7 Build the test plan**

Any project can be divided into units that can be further performed for detailed processing. Then a testing strategy for each of this unit is carried out. Unit testing helps to identity the possible bugs in the individual component, so the component that has bugs can be identified and can be rectified from errors

**CHAPTER 9**

**FUTURE ENHANCEMENT**

**9.1 FUTURE ENHANCEMENTS:**

For future enhancements to a healthcare optimization project focused on predicting patient test results, several avenues can be explored. This includes integrating advanced machine learning models like neural networks to capture intricate data patterns, implementing ensemble methods for improved prediction accuracy, and developing real-time data processing capabilities for responsive interventions. Dynamic feature engineering can automate feature extraction, while interactive visualization tools empower practitioners to analyze data effectively. Enhanced security measures ensure patient privacy, and integration with EHR systems leverages comprehensive patient histories. Continuous model monitoring and updates maintain accuracy, while scaling the service broadens its impact. Ongoing research drives innovation, keeping the project at the forefront of healthcare predictive modeling. These enhancements collectively optimize healthcare strategies and improve patient outcomes.

**CHAPTER 10**

**CONCLUSIONAND REFERENCES**

**10.1 CONCLUSION**

The healthcare optimization project leveraging machine learning for predicting patient test results aims to enhance healthcare strategies, personalize interventions, and improve patient outcomes. By utilizing Python libraries and web frameworks, the project facilitates efficient data analysis, modeling, and deployment of interactive applications for healthcare practitioners. Future enhancements including advanced models and real-time capabilities promise to further elevate impact and scalability, ultimately empowering professionals with actionable insights to optimize patient care and resource allocation in healthcare systems.

**10.2 REFERENCES**

Abul-Husn NS, Kenny EE (2019) Personalized medicine and the power of electronic health

records. Cell 177:58–69

Ahmad OF, Soares AS, Mazomenos E, Brandao P, Vega R, Seward E, Stoyanov D, Chand M,

Lovat LB (2019) Artificial intelligence and computer-aided diagnosis in colonoscopy: current

evidence and future directions. Lancet Gastroenterol Hepatol 4:71–80

Ardila D, Kiraly AP, Bharadwaj S, Choi B, Reicher JJ, Peng L, Tse D, Etemadi M, Ye W, Corrado

G, Naidich DP, Shetty S (2019) End-to-end lung cancer screening with three-dimensional deep

learning on low-dose chest computed tomography. Nat Med 25:954–961

Baiardini I, Heffler E (2019) Chapter 21 - The patient-centered decision system as per the 4Ps of

precision medicine. In: Agache I, Hellings P (eds) Implementing precision medicine in best

practices of chronic airway diseases. Academic Press, London, pp 147–151

1 See generally, on the different applications of Machine Learning and AI, in this book A Oliveira

and M A T Figueiredo - Artificial intelligence - historical context and state of the art; I Trancoso,

N Mamede, B Martins, H S Pinto and R Ribeiro - The impact of language technologies in the legal

domain; J Gonçalves-Sá and F L Pinheiro - Societal Implications of Recommendation Systems

- A Technical Perspective; A T Freitas - Data-driven approaches in healthcare - challenges and

emerging trends; M Correia and L Rodrigues - Security and Privacy; E Magrani and P G F Silva

- The Ethical and Legal Challenges of Recommender Systems Driven by Artificial Intelligence;

M Lanz and S Mijic - Risks associated with the use of natural language generation - Swiss civil

liability law perspective; M S Fernandes and J R Goldim - Artificial Intelligence and Decision

Making in Health - Risks and Opportunities; W Gravett - Judicial Decision-making in the Age

of Artificial Intelligence; D Durães, P M Freitas and P Novais - The Relevance of Deepfakes in

the Administration of Criminal Justice. See also, on Ethics, in this book P U Lima and A Paiva -

Autonomous and Intelligent Robots - Social, Legal and Ethical Issues; M C Patrão Neves and A

B Almeida - Before and Beyond Artificial Intelligence - Opportunities and Challenges; E Magrani

and P G F Silva - The Ethical and Legal Challenges of Recommender Systems Driven by Artificial

Intelligence; M S Fernandes and J R Goldim - Artificial Intelligence and Decision Making in

Health - Risks and Opportunities; M N Duffourc and D S Giovanniello - The Autonomous AI

Physician - Medical Ethics and Legal Liability; R Nogaroli and J L M Faleiros Júnior - Ethical

challenges of artificial intelligence in medicine and the triple semantic dimensions of algorithmic

opacity with its repercussions to patient consent and medical liability; and B A Ribeiro, H Coelho,

A E Ferreira and J Branquinho - Metacognition, Accountability and Legal Personhood of AI. See

finally, on AI and Healthcare, in this book M S Fernandes and J R Goldim - Artificial Intelligence

and Decision Making in Health - Risks and Opportunities; M N Duffourc and D S Giovanniello -

The Autonomous AI Physician - Medical Ethics and Legal Liability; and R Nogaroli and J L M

Faleiros Júnior - Ethical challenges of artificial intelligence in medicine and the triple semantic

dimensions of algorithmic opacity with its repercussions to patient consent and medical liability.

: 10.3390/s2014390 Bellini V, Guzzon M, Bigliardi B, Mordonini M, Filippelli S, Bignami E (2019) Artificial

intelligence: a new tool in operating room management. Role of machine learning models in

operating room optimization. J Med Syst 44:20

Carra G, Salluh JIF, Ramos FJDS, Meyfroidt G (2020) Data-driven ICU management: using big

data and algorithms to improve outcomes. J Crit Care 60:300–304

CBInsights (2021a) Here are the most active healthcare investors. https://www.cbinsights.com/

research/most-active-healthcare-vcs/. Accessed 8 Jan 2022

CBInsights (2021b) State of healthcare Q1’21 report: investment & sector trends to watch. https://

www.cbinsights.com/research/report/healthcare-trends-q1-2021/. Accessed 8 Jan 2022

Centers for Disease Control and Prevention (2018) Health insurance portability and accountability

act of 1996 (HIPAA). https://www.cdc.gov/phlp/publications/topic/hipaa.html. Accessed 17

Dec 2021

Dankar FK, Gergely M, Malin B, Badji R, Dankar SK, Shuaib K (2020) Dynamic-informed

consent: a potential solution for ethical dilemmas in population sequencing initiatives. Comput

Struct Biotechnol J 18:913–921

Dias R, Torkamani A (2019) Artificial intelligence in clinical and genomic diagnostics. Genome

Med 11:70

Elwyn G, Nelson E, Hager A, Price A (2020) Coproduction: when users define quality. BMJ Qual

Saf 29:711–716

European Commission (2020) WHITE PAPER on artificial intelligence-a European approach

to excellence and trust. https://ec.europa.eu/info/sites/default/files/commission-white-paperartificial-intelligence-feb2020\_en.pdf. Accessed 16 Nov 2021

European Commission (2021) Proposal for a regulation of the European parliament and of the

council. Laying down harmonised rules on artificial intelligence (Artificial intelligence act)

and amending certain union legislative acts. https://bit.ly/3AcDsCa. Accessed 16 Nov 2021

Fierce Healthcare (2021) Global investment in telehealth, artificial intelligence hits a new high in

Q1 2021. https://bit.ly/3qtk4hQ. Accessed 12 Nov 2021

Gerke S, Minssen T, Cohen G (2020) Ethical and legal challenges of artificial intelligence-driven

healthcare. Artif Intell Healthc:295–336. https://doi.org/10.1016/B1978-1010-1012-818438-

818437.800012-818435

Gerteis M, Edgman-Levitan S, Daley J, Delbanco TL (1993) Through the patient’s eyes:

understanding and promoting patient-centered care. Jossey-Bass, San Francisco, CA

Gijsberts CM, Groenewegen KA, Hoefer IE, Eijkemans MJC, Asselbergs FW, Anderson TJ,

Britton AR, Dekker JM, Engström G, Evans GW, de Graaf J, Grobbee DE, Hedblad B,

Holewijn S, Ikeda A, Kitagawa K, Kitamura A, de Kleijn DPV, Lonn EM, Lorenz MW,

Mathiesen EB, Nijpels G, Okazaki S, O’Leary DH, Pasterkamp G, Peters SAE, Polak JF,

Price JF, Robertson C, Rembold CM, Rosvall M, Rundek T, Salonen JT, Sitzer M, Stehouwer

CDA, Bots ML, den Ruijter HM (2015) Race/Ethnic differences in the associations of the

Framingham risk factors with carotid IMT and cardiovascular events. PLoS One 10:e0132321

Goecks J, Jalili V, Heiser LM, Gray JW (2020) How machine learning will transform biomedicine.

Cell 181:92–101

Grossglauser M, Saner H (2014) Data-driven healthcare: from patterns to actions. Eur J Prev

Cardiol 21:14–17

Gutierrez G (2020) Artificial intelligence in the intensive care unit. Crit Care 24:101

Handelman GS, Kok HK, Chandra RV, Razavi AH, Lee MJ, Asadi H (2018) eDoctor: machine

learning and the future of medicine. J Int Med 284:603–619

Heijlen R, Crompvoets J (2021) Open health data: mapping the ecosystem. Digit Health

7:20552076211050167

Hood L, Heath JR, Phelps ME, Lin B (2004) Systems biology and new technologies enable

predictive and preventative medicine. Science 306:640–643

Hood L, Balling R, Auffray C (2012) Revolutionizing medicine in the twenty-first century through

systems approaches.

Rajkomar A, Hardt M, Howell MD, Corrado G, Chin MH (2018) Ensuring fairness in machine

learning to advance health equity. Ann Int Med 169:866–872

Sagner M, McNeil A, Puska P, Auffray C, Price ND, Hood L, Lavie CJ, Han ZG, Chen Z,

Brahmachari SK, McEwen BS, Soares MB, Balling R, Epel E, Arena R (2017) The P4 health

spectrum – a predictive, preventive, personalized and participatory continuum for promoting

healthspan. Prog Prev Med 2:e0002

Sanchez-Pinto LN, Luo Y, Churpek MM (2018) Big data and data science in critical care. Chest

154:1239–1248

Savadjiev P, Reinhold C, Martin D, Forghani R (2020) Knowledge based versus data based: a historical perspective on a continuum of methodologies for medical image analysis. Neuroimaging

Clin N Am 30:401–415

Tauginien

˙

e

L, Hummer P, Albert A, Cigarini A, Vohland K (2021) Ethical challenges and dynamic

informed consent. In: Vohland K, Land-Zandstra A, Ceccaroni L, Lemmens R, Perelló J, Ponti

M, Samson R, Wagenknecht K (eds) The science of citizen science. Springer International

Publishing, Cham, pp 397–416

Tseng EK, Hicks LK (2016) Value based care and patient-centered care: divergent or complementary? Curr Hematol Malig Rep 11:303–310

U.S. Food and Drug Administration (2021) FDA releases artificial intelligence/machine learning action plan. https://www.fda.gov/news-events/press-announcements/fda-releases-artificialintelligencemachine-learning-action-plan. Accessed 8 Oct 2021

van der Niet AG, Bleakley A (2021) Where medical education meets artificial intelligence: ‘Does

technology care?’ Med Educ 55:30–36.

van der Wouden CH, Cambon-Thomsen A, Cecchin E, Cheung KC, Dávila-Fajardo CL, Deneer

VH, Dolžan V, Ingelman-Sundberg M, Jönsson S, Karlsson MO, Kriek M, Mitropoulou C,

Patrinos GP, Pirmohamed M, Samwald M, Schaeffeler E, Schwab M, Steinberger D, Stingl J,

Sunder-Plassmann G, Toffoli G, Turner RM, van Rhenen MH, Swen JJ, Guchelaar HJ (2017)

CORRIGENDUM: implementing pharmacogenomics in Europe: design and implementation

strategy of the ubiquitous pharmacogenomics consortium. Clin Pharmacol Ther 102:152

Weston AD, Hood L (2004) Systems biology, proteomics, and the future of health care: toward

predictive, preventative, and personalized medicine. J Proteome Res 3:179–196

Wong-Lin K, McClean PL, McCombe N, Kaur D, Sanchez-Bornot JM, Gillespie P, Todd S, Finn

DP, Joshi A, Kane J, McGuinness B (2020) Shaping a data-driven era in dementia care pathway

through computational neurology approaches. BMC Med 18:398–398

Xu F, Uszkoreit H, Du Y, Fan W, Zhao D, Zhu J (2019) Explainable AI: a brief survey on history,

research areas, approaches and challenges. In: Tang J, Kan MY, Zhao D, Li S, Zan H (eds)

Natural language processing and Chinese computing NLPCC 2019 Lecture notes in computer

science. Springer International Publishing, Cham, pp 563–574