Document Information

Analyzed document	Diseases Prediction System Based on Symptom Using Machine Learning.pdf (D185230271)
Submitted	2024-02-04 13:00:00 UTC+01:00
Submitted by	
Submitter email	2022981099@student.uitm.edu.my
Similarity	5%
Analysis address	raseeda.UiTM@analysis.ouriginal.com

Sources included in the report

SA	**UNIVERSITI TEKNOLOGI MARA (UiTM) / CSP650-FullReport_MuhammadHafizBinMohd.Saufan (1).pdf** Document CSP650-FullReport_MuhammadHafizBinMohd.Saufan (1).pdf (D184813755) Submitted by: 2022923567@student.uitm.edu.my Receiver: samsi260.UiTM@analysis.ouriginal.com	3
SA	**UNIVERSITI TEKNOLOGI MARA (UiTM) / Proposal Report_Nur Alia Shakila_CS2305B.docx** Document Proposal Report_Nur Alia Shakila_CS2305B.docx (D125615167) Submitted by: 2019294656@student.uitm.edu.my Receiver: jamaluddinjasmis.UiTM@analysis.ouriginal.com	1
W	URL: https://www.ncbi.nlm.nih.gov/books/NBK499956/ Fetched: 2/4/2024 1:00:00 PM	1
SA	**rithikanew (4) (2).docx** Document rithikanew (4) (2).docx (D105034607)	1
SA	**UNIVERSITI TEKNOLOGI MARA (UiTM) / Amir Mustaqim Bin Aman_2022912795_CSP650 Final Report Submission.pdf** Document Amir Mustaqim Bin Aman_2022912795_CSP650 Final Report Submission.pdf (D184285955) Submitted by: 2022912795@student.uitm.edu.my Receiver: aliseman.UiTM@analysis.ouriginal.com	1
W	URL: https://doi.org/10.3390/ijerph17145218 Fetched: 2/4/2024 1:00:00 PM	1
SA	**report To check Pilgirism.pdf** Document report To check Pilgirism.pdf (D172727672)	1
SA	**UNIVERSITI TEKNOLOGI MARA (UiTM) / Batrisyia Ajuwa binti Rozaini_Final Report.pdf** Document Batrisyia Ajuwa binti Rozaini_Final Report.pdf (D158294162) Submitted by: 2021155845@student.uitm.edu.my Receiver: saleha.UiTM@analysis.ouriginal.com	1
W	URL: https://doi.org/10.52866/ijcsm.2022.01.01.013 Fetched: 2/4/2024 1:00:00 PM	1

W	URL: https://doi.org/10.35940/ijrte.E6482.018520 Fetched: 2/4/2024 1:00:00 PM	1
W	URL: https://doi.org/10.3991/ijim.v14i11.13269 Fetched: 2/4/2024 1:00:00 PM	1
SA	**UNIVERSITI TEKNOLOGI MARA (UiTM) / REPORT-2019294656.docx** Document REPORT-2019294656.docx (D142598150) Submitted by: 2019294656@student.uitm.edu.my Receiver: jamaluddinjasmis.UiTM@analysis.ouriginal.com	1
W	URL: https://doi.org/10.3390/jpm10020021 Fetched: 2/4/2024 1:00:00 PM	1
W	URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/ Fetched: 5/27/2022 7:08:38 AM	2
W	URL: https://doi.org/10.38094/jastt20179 Fetched: 2/4/2024 1:00:00 PM	3
W	URL: https://doi.org/10.9734/jpri/2021/v33i33b31810 Fetched: 2/4/2024 1:00:00 PM	1
W	URL: https://www.hindawi.com/journals/jhe/2021/9983652/ Fetched: 6/13/2021 2:40:40 PM	1
W	URL: https://www.researchgate.net/publication/361308218_An_Analysis_of_the_Impact_of_the_COVID-19_Pandemic_on_the_Economic_and_Health_Aspect_in_Indonesia Fetched: 1/5/2023 12:21:12 PM	1
W	URL: https://doi.org/10.2196/19713 Fetched: 2/4/2024 1:00:00 PM	1
W	URL: https://doi.org/10.1097/ACM.0000000000002518 Fetched: 2/4/2024 1:00:00 PM	1
SA	**UNIVERSITI TEKNOLOGI MARA (UiTM) / FINAL REPORT_MUHAMMAD HANIF BIN ADNAN_2022912847.pdf** Document FINAL REPORT_MUHAMMAD HANIF BIN ADNAN_2022912847.pdf (D184517383) Submitted by: 2022912847@student.uitm.edu.my Receiver: nasiroh.UiTM@analysis.ouriginal.com	1
W	URL: https://doi.org/10.1186/s12936-023-04446-0 Fetched: 2/4/2024 1:00:00 PM	1

Entire Document

UNIVERSITI TEKNOLOGI MARA Diseases Prediction Based on Symptoms Using Machine Learning Technique Mohamad Afham Aiman Bin Mohamad Asri

53%	MATCHING BLOCK 1/27	SA	CSP650-FullReport_MuhammadHafizBinMohd.Saufan ... (D184813755)

BACHELOR OF COMPUTER SCIENCE (HONS.) February 2024 ii Universiti Teknologi MARA Diseases Prediction Based on Symptoms Using Machine Learning Technique Mohamad Afham Aiman Bin Mohamad Asri Thesis submitted in fulfillment of the requirements for Bachelor of Computer Science (Hons.) College of Computing, Informatics and Media February 2024 iii SUPERVISOR'S APPROVAL Diseases Prediction Based on Symptoms Using Machine Learning Technique By MOHAMAD AFHAM AIMAN BIN MOHAMAD ASRI 2022981099 This thesis was prepared under the supervision of project supervisor, Dr. Firdaus Fadzil. It was submitted to the College of Computing, Informatics and Media and was accepted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Hons) Computer Science. Approved by ... ... Dr. Firdaus Fadzil Project Supervisor February, 1,2024 iv STUDENT'S DECLARATION I certify that this report and project to which it refers are the product of my own work and that any idea or quotation from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referring practices of the discipline. ... ...

MOHAMAD AFHAM AIMAN BIN MOHAMAD ASRI 2022981099 FEBRUARY, 2,2024
v ACKNOWLEDGEMENT I am grateful and offer praise and thanks to Allah for His immense power and abundant blessings, as I have successfully completed this research within the given timeframe. I extend my heartfelt appreciation to my supervisor, Dr Ahmad Firdaus bin Ahmad Fadzil, for generously providing his time, guidance, and patience, which proved instrumental in obtaining valuable information for the completion of this work. Furthermore, I would like to express my sincere gratitude and deep appreciation to Dr. Raseeda Binti Hamzah, my lecturer for the CSP650 subject project. Her advice and guidance throughout the semester were invaluable in preparing this thesis. I would also like to acknowledge and thank my dear friends who stood by me during challenging times, providing valuable ideas and solutions. Despite our busy schedules, we have always been there for one another. May Allah bestow His blessings upon all of you. Last but not least, I want to thank me. I want thank me for believing in me. I want to thank me for doing all this hard work. I want to thank me for having no days off. I want to thank me for never quitting.
vi TABLE OF CONTENTS CONTENT PAGE SUPERVISOR'S APPROVAL iii STUDENT'S DECLARATION iv ACKNOWLEDGMENT v

43%	MATCHING BLOCK 2/27	SA	Proposal Report_Nur Alia Shakila_CS2305B.docx (D125615167)

TABLE OF CONTENTS vi LIST OF TABLES ix LIST OF FIGURES x LIST OF ABBREVIATIONS xi CHAPTER 1: INTRODUCTION 1 1.1 Background of Study 2 1.2 Problem Statement 3 1.3 Project Aim 3 1.4 Project Objective 3 1.5 Project Scope 3 1.6 Project Significance 3 1.7 Project Outline 4 CHAPTER 2:

LITERATURE RIVIEW 5 2.1 Symptoms and Diseases 5 2.2 Traditional Diseases Diagnosis 7 2.3 Automated Prediction Techniques 8 2.4 Machine Learning Techniques 9 2.5 Software Development Model 13 2.6 Similar Work 15 2.6.1 Mayo Clinic Symptom Checker 15 2.6.2 Symptomate 16 2.6.3 WebMD Symptom Checker 17 2.6.4 Comparison Between Similar Projects 17 2.7 Summary 18
vii CHAPTER 3: METHODOLOGY 19 3.1 Software Development Methodology 19 3.2 Project Framework 21 3.3 Objective 1 23 3.4 Objective 2 23 3.4.1 Data Collection 24 3.4.2 System Design 26 3.4.3 Machine Learning Design 27 3.4.4 System UI Design 30 3.4.5 System Implementation 32 3.4.5.1 Hardware Requirement 32 3.4.5.2 Software Requirement 33 3.5 Objective 3 34 3.5.1 System Testing 35 CHAPTER 4: Design and Development 37 4.1 Data Pre-Processing 37 4.2 Random Forest Model 41 4.2.1 Dictionary Setup 41 4.2.2 Features Extraction 42 4.2.3 Build Model 43 4.2.4 Model Training 44 4.2.5 Model Testing 45 4.3 Web Building 46 4.3.1 Web Integration 46 4.3.2 Web Development 48 4.3.2.1 Diagnosis Page 48 4.3.2.2 Result Page 50 4.3.2.3 Javascripts 51 4.4 Summary 53 CHAPTER 5: Result and Discussion 54 5.1 Functionality Testing 54
viii 5.1.1 Test Case 1 56 5.1.2 Test Case 2 57 5.1.3 Test Case 3 58 5.1.4 Test Case 4 59 5.1.5 Test Case 5 60 5.1.6 Test Case 6 61 5.1.7 Test Case 7 62 5.1.8 Test Case 8 63 5.1.9 Test Case 9 64 5.1.10 Test Case 10 65 5.2 Summary of Test Cases 66 5.3 Conclusion 66 CHAPTER 6: Conclusion and Recommendation 67 6.1 Objective 67 6.1.1 Objective 1 67 6.1.2 Objective 2 68 6.1.3 Objective 3 68 6.2 Limitation 68 6.3 Future Recommendation 69 REFERENCES 70 APPENDICES 73 APPENDIX A GANT CHARTS 74
ix LIST OF TABLES TABLE PAGE 2.1 Supervised vs Unsupervised Learning 10 2.2 Machine Learning Algorithm 11 2.3 Similar Work 18 3.1 Hardware Requirement 32 3.2 Software Requirement 33 3.3 User Acceptance Form 35 5.1 Functionality Test 54 5.2 Test Case 1 56 5.3 Test Case 2 57 5.4 Test Case 3 58 5.5 Test Case 4 59 5.6 Test Case 5 60 5.7 Test Case 6 61 5.8 Test Case 7 62 5.9 Test Case 8 63 5.10 Test Case 9 64 5.11 Test Case 10 65 5.12 Summary of Test Cases 66

xii LIST OF ABBREVIATIONS k-NN k-Nearest Neighbours CNN Convolutional Neural Network SVM Support Vector Machine OR Odd Ratio CI Confidence Interval
1 CHAPTER 1 INTRODUCTION This chapter will provide the introduction as well as the background of the project being conducted. This chapter will consist of the background of the study, problem statement, aim of study, the objectives, scope and significance project about health diagnosis system. 1.1 Background of Study Health diagnosis is the process of identifying a health condition or disease based on an evaluation of a patient's symptoms, medical history, and diagnostic tests. Many lives can be saved with quick data analysis and accurate disease prediction from symptoms. Early disease detection enables doctors to prescribe effective treatments (Ali & Divya, 2020). This project will be focusing on digital health diagnosis which is an emerging field that utilizes technology to identify and diagnose health conditions based on a patient's symptoms. This approach combines traditional diagnostic methods with the power of digital technology to provide more accurate and efficient diagnoses by utilizing data mining and machine learning techniques. Artificial intelligence-powered digital symptom assessment tools could assist patients in finding the appropriate level of urgency and type of care (Miller et al., 2020). However, this process can be complex and may involve a variety of medical professionals, such as physicians, nurses, and specialists. Patients need a quick way to diagnose their problem which will make diagnosis questioning shorter and cause the result to be less accurate. Thus, an interactive system with better visualization can improve the success of the health diagnosis. The primary goal of developing this project was to enable users to check their health while relaxing in a convenient location (Kamble et al., 2021).
2 1.2 Problem Statement Digital diagnosis systems can help reduce healthcare costs by streamlining the diagnostic process and reducing the need for unnecessary tests and procedures. By providing a more accurate and efficient diagnosis, healthcare providers can avoid unnecessary expenses and improve the cost-effectiveness of healthcare delivery. Focus has switched to the digital health community in this period of extreme medical crisis in order to provide prospective health solutions to decrease the impact of epidemic (Kapoor et al., 2020). Compared to conventional methods of health diagnosis, it can provide disease prediction with a higher degree of accuracy based on user symptoms. However, effective early stage diagnosis is still seen as a poorly posed challenge (Kumar et al., 2021). Up to 70% of medical mistakes are the result of diagnostic errors (Royce et al., 2019). It is more difficult to prevent diagnostic errors than it is to incorporate safety measures into healthcare systems (Royce et al., 2019). The examination process takes time and is subject to human mistake (Yu et al., 2023). This can be caused by patients that have difficulty communicating their symptoms or healthcare providers may have hindsight bias in diagnosis of diseases. It is a complex problem since

52%	MATCHING BLOCK 3/27	W

it is difficult to identify a consistent source of errors and to offer a consistent workable solution that reduces the likelihood of a recurrent event (

Rodziewicz & Hipskind., 2019). In conclusion, the lack of an interactive system to accurately diagnose diseases, which can delay the treatment and be time-consuming. Therefore, an interactive system will be helpful in providing a better diagnosis process. The integration between computer technology and conventional diagnosis methods has proved it is hoped that by proposing this project, it should be able to simplify the difficulty of medical diagnosis and decrease medical diagnosis error.
3 1.3
Project aim

100%	MATCHING BLOCK 4/27	SA	rithikanew (4) (2).docx (D105034607)

The aim of this project is to develop a system

that
is able to list all potential diseases based on the symptoms input by the user. 1.4 Objectives This project intends to create a system to forecast diseases based on symptoms in order to overcome incorrect health diagnoses and save time during the medical examination procedure. Consequently, the goals also include: 1. To identify suitable techniques and algorithms in recognizing potential diseases based on symptoms input. 2. To develop a web-based system that aligns with the technique and algorithm used. 3. To test the functionality and confidence level of the developed system. 1.5 Scope of project This project's data visualization focuses on disease's symptoms and is based on a dataset made available on the Kaggle website from Pranay Patil. 1.6 Significance A project on digital health diagnosis systems has a lot of promise to boost accessibility, results, and healthcare delivery. Healthcare providers can better serve patients, lower healthcare costs, and enhance public health by creating and deploying efficient digital diagnosis systems.
4 1.7 Outline of project The project outline as follows: Chapter 1: This chapter discussed a background overview of the project as well as the justification for the system's development.

62%	MATCHING BLOCK 5/27	SA	Amir Mustaqim Bin Aman_2022912795_CSP650 Final ... (D184285955)

The background of the study, problem statement, aim, objectives, significance, scopes and outline of the project

are all included in this chapter. Chapter 2: This chapter discussed the project's literature review, which will include descriptions of the research conducted on the project's issue. This chapter also includes comparisons of existing apps and approaches that can be implemented. Chapter 3: The methodology that will be used throughout the project will be explained in this chapter. This chapter includes an explanation of the project's various plans and phases. Chapter 4: The implementation that will be done throughout the project will be explained in this chapter. This chapter includes an explanation of the project's data preparation, random forest model, and web building. Chapter 5: The result and discussion that collected throughout the project will be explained in this chapter. This chapter includes an explanation of the functionality test. Chapter 6: The overall conclusion for each objective and recommendation on the project will be explained in this chapter. This chapter includes the conclusion summary, limitation and future recommendation.
5 CHAPTER 2 LITERATURE REVIEW This chapter provides a literature review that has been compiled from various articles related to diseases, symptoms, automated prediction techniques, and machine learning techniques. The literature review involves a comprehensive analysis of the existing literature on these topics, with the aim of identifying the current state of knowledge, identifying gaps in knowledge, and determining the direction and scope of the study. In particular, this chapter focuses on the use of automated prediction techniques and machine learning techniques in the context of disease diagnosis. The literature review explores various approaches to automated disease prediction, including the variety of machine learning algorithms. By analyzing and synthesizing existing literature, this chapter aims to provide a comprehensive understanding of the current state of research in the area of disease prediction using machine learning techniques. 2.1 Symptoms and Diseases People today deal with a variety of ailments as a result of their lifestyle choices and the surroundings. An abnormal condition known as a disease is one that specifically damages an organism's structure or function, either completely or in part, without being instantly triggered by any external shock. It is widely accepted that illnesses are classified as diseases if they have recognized signs and symptoms. While symptoms are an individuals' subjective reactions to sickness or damage are known as symptoms. These can include emotional states like anxiety or depression as well as bodily feelings like pain, nausea, or weariness. Understanding symptoms is essential for accurate medical condition diagnosis and therapy. Both symptoms and diseases are interrelated components of human health. Diseases are illnesses that impair the body's ability to operate normally and can impact a variety of organs and systems. The patient's subjective reactions to the condition are referred to as symptoms. Pain, fever, exhaustion, and other bodily and mental manifestations are examples of symptoms. Although gastrointestinal symptoms are very common, many people who experience
6 them have no natural explanation for them. The majority of these individuals will be classified as having a functional gastrointestinal disorder, such as functional constipation, functional dyspepsia, or irritable bowel syndrome. At any given time, these illnesses can impact up to 40% of people, and two-thirds of those individuals will experience persistent symptoms that change over time (Black et al., 2020). The odds ratio (OR) for

58%	MATCHING BLOCK 6/27	W

anosmia and dysgeusia in COVID-19 cases was reported to be 100.7 (95% Confidence Interval [CI] = 26.5-382.6) and OR = 51.8 (95%CI 16.6- 161.9), respectively, as of the early year 2020, when the world had been infected by Covid-19 and 29.3% of the cases never showed any symptoms with a significant increase in the estimated risk, anxiety was reported by 16.6% of COVID-19 patients and depression by 20.3% of cases (OR = 4.3; 95%CI = 2.4-7.4 for anxiety, OR = 3.5; 95%CI = 2.0-6.0 for depression) (Magnavita et al., 2020). This medical diagnosis result has been analyzed to find the relationship to determine early signs of infectious virus. Early identification of infected individuals allows for timely implementation of isolation measures and contact tracing, which can significantly reduce the transmission of the virus. Therefore, accurate and fast diagnosis plays a critical role in the effective management and control of infectious diseases. In this era of the AI world, the use of automated prediction techniques can significantly increase the efficiency and accuracy of various processes, including disease diagnosis which will be discussed in further details in chapter 2.2.

7 2.2 Traditional Diseases Diagnosis Despite the development of cutting-edge technologies and diagnostic techniques, classical disease diagnosis still has a lot of importance in the quickly developing area of medicine. Despite the fact that new procedures have transformed medical diagnostics, it is critical to acknowledge the ongoing significance of conventional diagnostic approaches in healthcare. A thorough physical examination and the collection of a patient's clinical history are crucial steps in the traditional illness diagnosis process (Zhang et al., 2020). Skilled healthcare practitioners can acquire vital information about symptoms, their duration, and any connected circumstances by carefully listening to the patient's story and carefully evaluating their body. Physical examination demonstrated an average diagnostic accuracy [area under the curve (AUC) = 0.76] with a pooled sensitivity and specificity of 71% and 69%, respectively (Zhang et al., 2020). This procedure lays the groundwork for more research and directs following diagnostic procedures. The process of pattern recognition is also used in traditional illness diagnosis, in which medical practitioners rely on their substantial clinical expertise to pinpoint distinctive patterns of symptoms and signs connected to certain diseases (Chen & Jain, n.d.). Clinicians gain an instinctive capacity to identify these patterns via years of training and exposure to a variety of situations, assisting in precise diagnosis and treatment recommendations. Traditional disease diagnosis is still a vital component of healthcare even as contemporary diagnostic tools advance. Traditional approaches help with accurate diagnosis and individualized patient care by emphasizing clinical history, physical examination, pattern identification, and differential diagnosis. The advantages of both conventional and contemporary diagnostic methods can be combined to increase diagnostic precision, boost patient outcomes, and guarantee holistic healthcare delivery (Ibrahim & Abdulazeez, 2021). The science of disease diagnosis will surely advance as a result of embracing the symbiotic relationship between tradition and innovation, which will be advantageous to both patients and healthcare professionals (Abdar et al., 2020). The integration of traditional disease diagnosis with automated prediction techniques paves the way for a comprehensive and advanced approach to healthcare. This will be further discussed in detail on chapter 2.3.

8 2.3 Automated Prediction Techniques The accurate and timely diagnosis of various diseases is one of the key challenges facing healthcare research (Abdar et al., 2020). Due to its capacity to quickly and effectively analyze enormous volumes of data, automated prediction approaches have gained popularity in recent years. These methods make use of statistical models and machine learning algorithms to forecast trends and outcomes based on previous data. They can be used in a variety of industries, including marketing, healthcare, finance, and sports analytics. Automated prediction algorithms, for instance, can be used to manage risk, forecast stock prices, and find trading opportunities. Nowadays, the analysis of data based on prior data and experience, speech and facial recognition, weather forecasting, and medical diagnostics are just a few of the many applications that are already available (Jha et al., 2019). By enabling data-driven decision-making and enhancing overall efficiency, automated prediction techniques have completely changed how businesses and organizations work. One of the areas where automated prediction techniques are useful is in weather forecasting. Various sensors are used to collect real-time data on temperature, humidity, and pressure in order to estimate the likelihood of rain (Mohan et al., n.d.). This can be critical for emergency response teams and other organizations that need to prepare for and respond to severe weather. Even in the crypto world the automated prediction technique surely gives a huge advantage to those that invest in cryptocurrency. Stock brokers and cryptocurrency traders may have an advantage in the market thanks to price prediction (Poongodi et al., 2020). With the help of machine learning the healthcare expertise has been greatly assisted in diagnosis and management areas using prediction techniques. These methods have been used in the treatment, prognosis, and also diagnosis of ailments (Rong et al., 2020). Other than that, automated prediction techniques are also useful in marketing. Financial time series are challenging to anticipate because the prices of financial assets are non-linear, dynamic, and unpredictable (Henrique et al., 2019). By analyzing consumer data and behavior, these techniques can help predict which products or services are likely to be most successful in the market. This can help businesses make more informed decisions about product development, marketing campaigns, and pricing strategies. In conclusion, automated prediction techniques are becoming increasingly important in many different areas, and are helping to improve

9 decision-making and outcomes in finance, healthcare, weather forecasting, marketing, and many other fields. As these techniques continue to evolve and improve, they are likely to become even more valuable in the years ahead, as machine learning techniques become increasingly relied upon in the future. Machine learning techniques have in depth detail that need to be understood which will be discussed on chapter 2.4. 2.4 Machine Learning Techniques Machine learning is a branch of artificial intelligence that involves training computer systems to learn from data and make predictions or decisions without explicit programming (Aggarwal et al., 2022). There are different types of machine learning algorithms, each with its own characteristics and applications such as supervised learning and unsupervised learning which have their own type of algorithm. Figure 2.1 shows examples of supervised and unsupervised learning algorithms. Supervised learning and unsupervised learning are two fundamental branches of machine learning that differ in their approaches: supervised learning involves training models using labeled data, where the desired output is known, enabling the model to make predictions or classifications, while unsupervised learning focuses on analyzing unlabeled data to discover patterns, structures, or relationships without specific guidance, allowing for exploratory data analysis and the identification of hidden insights that can aid in data understanding, clustering, or anomaly detection. Table 2.1 shows the comparison between these types of machine learning.
10 Figure 2.1 example of supervised vs unsupervised learning (Source: Smriti, 2021) Table 2.1 Supervised vs Unsupervised learning Supervised Learning Unsupervised Learning Training Data Labeled data where each data point has a known target value Unlabeled data without known target values Objective Predicting or classifying target variables Discovering patterns, structures, or relationships Example Algorithms Decision trees, support vector machines, neural networks Clustering algorithms, dimensionality reduction techniques Data Requirement Requires labeled data for training Can work with unlabeled data or a mix of labeled/unlabeled data Evaluation Accuracy, precision, recall, F1-score, etc. Cohesion, separation, silhouette score, etc. Applicability Predictive modeling, classification, regression Anomaly detection, exploratory data analysis Use Case Examples Spam email classification, sentiment analysis Customer segmentation, market basket analysis Machine learning techniques are a subset of artificial intelligence that allow computer systems to learn from data and improve their performance over time (Battineni et al., 2020) . Machine learning is concerned with building algorithms that can automatically identify patterns in data and make predictions based on those patterns. These techniques are used in a wide variety of applications, including finance, healthcare,
11 marketing, and many others. Machine learning's fundamental premise is to teach a computer system to spot patterns in data. Usually, a lot of data is fed into the system, and an algorithm is used to find patterns and predict the future (Quer et al., 2021). The algorithm becomes stronger at seeing patterns and making precise predictions as more data is fed into it. Research have shown that precision and accuracy also effected by the algorithm technique that being implemented. Table 2.2 show that machine learning algorithm and the conclusion that have been made by others researcher. Table 2.2 Machine Learning Algorithm comparison No. Author Machine Learning Algorithm Conclusion 1 (Islam et al., 2020) CNN CNN=94% 2 (Banerjee, 2020) SVM, Random Forest, ANN algorithms Accuracy Random Forest =70%, SVM=80% and ANN=96% 3 (Sri Eshwar College of Engineering & Institute of Electrical and Electronics Engineers, 2020) Random Forest, Decision Tree, Logistic Regression, SVM, and KNN algorithms Accuracy Random Forest =85.71%, decision tree =74.28%, Logistic Regression=74.28%, 4 (Islam Trishna et al., 2019) KNN, random forest, and naïve Bayes algorithm Accuracy KNN=95.8%, random forest=98.6% and naïve Bayes =93.2% 5 (Institute of Electrical and Electronics Engineers, 2019) KNN, Random Forest, Logistic Regression, and Ensemble Method algorithms Accuracy KNN=87%, Random Forest= 87%, Logistic Regression=87%, Hard Voting Ensemble Method=90% Based on their traits and uses, machine learning algorithms can be divided into numerous different types (Chauhan et al., 2021). For prediction and classification
12 problems, supervised learning algorithms, such as decision trees, support vector machines, and neural networks, learn from labeled training data. Unsupervised learning algorithms, such as dimensionality reduction methods and clustering algorithms, search for patterns or groupings in unlabeled data. There is a wide variety of algorithms available for both supervised learning, such as decision trees, support vector machines, and neural networks, which enable prediction and classification based on labeled data, as well as unsupervised learning, encompassing clustering algorithms, dimensionality reduction techniques, and association rule learning, which help to identify patterns, structures, and relationships in unlabeled data for exploratory analysis and data understanding. Figure 2.2 shows examples of algorithms for both supervised and unsupervised learning. While machine learning plays an important role in this research, software development models also contribute an important role in this research which will further discuss in chapter 2.5. Figure 2.2 example of algorithm for supervised and unsupervised learning (Source: Suryakanthi,2020)
13 2.5 Software Development Model Choosing the best software development model is essential in the quickly changing world of software development if you want to produce high-quality software solutions and complete projects successfully (Angela Adanna & Francisca Nonyelum, 2020). The selection of a development model offers a well-organized framework that directs all phases of software development, including gathering requirements, designing, implementing, testing, deploying, and maintaining new versions of existing software (Lalband & Kavitha, 2019). It creates the conditions for productive teamwork, successful processes, and the accomplishment of project objectives. Understanding and utilizing the various software development models becomes crucial as organizations work to create software products that match customer expectations, stick to project timeframes, and stay inside budgetary limitations (Ramirez et al., 2020). To help software development teams and project stakeholders make wise decisions and achieve successful project outcomes, this article delves into the subtleties of various software development models. It does so by highlighting their distinctive qualities, advantages, and best-suited scenarios. We can learn more about how these models influence the software development process, encourage productive communication and teamwork, and ultimately help to produce solid and dependable software solutions by examining their problem. One of the best software development model is shown in Figure 2.3.

14 Figure 2.3 Waterfall Model (Source: Arslan,2023) A conventional, linear method that employs a sequential flow is the waterfall model. It has different stages, such as gathering requirements, designing, implementing, testing, deploying, and maintaining. This paradigm works best for projects with clear requirements that don't change much throughout the development process. It is strictly sequential nature guarantees distinct milestones and makes managing and documenting each phase easier. The waterfall model is a life cycle paradigm for software development that was first introduced by Royce in the 1970s (Aroral, n.d.). Projects used the waterfall method of software development prior to the advent of agile software development. The waterfall approach resembled a succession of logical periods in which development used to occur in a linear way.

15 2.6 Similar Work There are some systems for predicting diseases that have been developed for use in disease diagnosis. Based on that, we evaluate three projects that are comparable in order to assist us create a method for predicting diseases for this project. 2.6.1 Mayo Clinic Symptom Checker The renowned Mayo Clinic, a non-profits medical center and research organization with headquarters in the United States, offers the Mayo Clinic Symptom Checker as an online resource. The purpose of the Symptom Checker is to help users determine the reasons for their symptoms and offer broad advice on when to consult a doctor. Sophisticated AI engine that analyzes user-reported symptoms and offers individualized health advice using a tremendous quantity of medical data and specialist knowledge. The method takes into account a number of variables, including the user's unique symptoms as well as age, gender, and medical history. Users can enter their symptoms into the Mayo Clinic Symptom Checker's user-friendly interface by choosing from a long list or conducting a free-text search. Users may enter several symptoms and add further information, such as the severity and duration of symptoms. The Symptom Checker creates a list of potential diseases or causes that might be related to the reported symptoms based on the information given. A brief description and supplementary resources are provided with each potential diagnosis for further research.

16 2.6.2 Symptomate App Leading this healthcare shift is the AI-powered web application Symptomate. Symptomate is a tool that helps users evaluate their symptoms and identify possible explanations, giving them the power to decide on their own health. We can examine the characteristics, advantages, and effects of Symptomate as a personal symptom assessment companion in this section. The innovative symptom checker Symptomate stands out for using artificial intelligence to analyze and assess reported symptoms. Through an easy-to-use interface, users can input their symptoms, and a sophisticated algorithm in the app compares the information to a large medical database. A thorough assessment that includes probable conditions and pertinent medical data is the outcome.

17 2.6.3 WebMD Symptom Checker A well-known online tool offered by WebMD, a dependable provider of health-related resources and information, is the WebMD Symptom Checker. With the aid of the Symptom Checker, people can learn more about their health issues and find out the reasons or causes that have possibility based on their symptoms. A user-friendly interface on the WebMD Symptom Checker enables users to enter their symptoms by choosing from a long list or by inputting them in free-text style. Users may enter several symptoms and give details about their timing, severity, and any coexisting conditions. After the symptoms are entered, the Symptom Checker analyzes the data and creates a list of potential diseases or disorders that might be the source of the reported symptoms. A description of each possible diagnosis is provided, along with information on typical symptoms and potential causative factors. 2.6.4 Comparison Between Similar Projects Based on the information provided in sections 2.6.1, 2.6.2, and 2.6.3, it is clear that there are numerous potential approaches to creating a useful system for predicting diseases. While the Mayo Clinic Symptom Checker is a valuable tool, it can be improvised by enhancing the UI design for it to be much more user friendly. When it comes to interactive diagnosis, Symptomate excels at keeping the query straightforward and user-friendly, allowing them to choose from a list of options or by providing word suggestions. To elicit more specific information regarding the symptoms, we can benefit from including contextual queries. This can aid in gathering more pertinent information and offer a more precise evaluation of the user's condition. The WebMD Symptom Checker, on the other hand, is user-friendly and interactive. However, it will improve the system by implementing a significant comparing with mild ailment distinction. User can priorities their health issues by creating a better distinction between urgent diseases that demand immediate medical attention and lesser ailments. By doing so, users may experience less unwarranted fear and receive better advice on when to seek emergency medical attention.

18 Table 2.3 Similar Work Mayo Clinic Symptom Checker Symptomate The WebMD Symptom Checker Proposed system User friendly interface No Yes Yes Yes Body map No Yes Yes Yes Confidence level of diseases diagnosis No No Yes Yes Severity level No No No Yes Diseases precautions Yes Yes Yes Yes 2.7 Summary Based on the research on journal, article and book that have been conducted, The algorithms that are suitable to be used in developing the prediction system are Random Forest algorithm and KNN algorithm. KNN, the k-Nearest Neighbors (k-NN) algorithm and Random Forest algorithm

100%	**MATCHING BLOCK 7/27**	SA	report To check Pilgirism.pdf (D172727672)

is a popular machine learning algorithm used for both classification and regression tasks, which is highly suitable for a diseases prediction system as it can effectively handle complex medical data, handle missing values, and provide accurate predictions by combining multiple decision trees to mitigate over fitting and improve the robustness of the predictive models. After conducting literature review, the literature review on developing a disease prediction system has provided valuable insights into the various aspects of this field. The review has highlighted the significance of utilizing machine learning techniques, such as supervised learning algorithms, in accurately predicting and diagnosing diseases Overall, this literature review serves as a foundation for future research and development efforts in creating effective and efficient disease prediction systems that can revolutionize healthcare delivery and improve patient care.

19 CHAPTER 3 METHODOLOGY This chapter explains how the project technique progressed from the initial phase to the final project report. Here, the whole methodology of the study including the chosen approach and its execution is described. Research technique is a methodical approach to addressing the research problem by ostensibly incorporating many advancements. Methodology includes both the process itself and the final system. 3.1 Software Development Methodology Requirement collection and analysis, system design, implementation, testing, deployment, and maintenance are the usual steps that make up the Waterfall model. Each of these phases is carried out in exact order, building on the results of the one before it. This sequential method enables distinct deliverable, explicit milestones, and thorough documentation at each level of the development cycle. Figure 3.1 shows the illustration flow of

96%	MATCHING BLOCK 8/27	SA	Batrisyia Ajuwa binti Rozaini_Final Report.pdf (D158294162)

waterfall model. Figure 3.1 Waterfall Model Illustration 20 (Source: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)

When compared to other software development methodologies like Agile, Spiral, or hybrid models that allow for flexibility, iteration, and early user interaction, the Waterfall model may not be the best option for creating diseases prediction systems. These methodologies are more effective in this field. These methods are better able to adapt to changing needs, take into account user feedback, and take into account the dynamic nature of disease prediction systems.

21 3.2 Project Framework An organized method used in software development projects, the Software Development Life Cycle (SDLC) serves as a road map or guideline for the entire procedure, from the initial software concept to its deployment and ongoing maintenance. The Waterfall model is an established and systematic SDLC method that progresses linearly through various phases. In order to best satisfy the project requirements, the current system will be improved and modified using the aforementioned paradigm. Since each phase must be completed before moving on to the next, there is no phase overlap in the waterfall development model. Requirement analysis, design, implementation, testing, and assessment are the five main phases of the waterfall development methodology. Figure 3.2 show the overall process in developing this system

22 Figure 3.2 Framework for diseases prediction system

23 3.3 Objective 1 The primary discussion on this subject will be focused on phase 1 requirement analysis. This will summarize all information from chapter 1. Analysis of requirements is the first step in designing a waterfall model. It is clear from reading the essay on the machine learning-based illnesses prediction system that each model has advantages and disadvantages. There are ways to identify diseases based on symptoms, which are the initial signs of illnesses and the body's reactions to an unknown substance in a body system, according to past studies that have been published in study publications. The article on earlier studies also illustrates which models are more effective at predicting diseases. After recognizing and understanding the requirements for a disease prediction system, the Random Forest based on research findings, is the suitable technique to create the system since it has the average of high accuracy compared to the other machine learning model. 3.4 Objective 2 This chapter focuses on the data collection, designing and implementation phases of the creation of the disease prediction system. As for the proposed project's datasets it came from the Kaggle website, which includes a variety of diseases and symptoms. The necessary datasets are signs and symptoms of the illnesses. In addition to that dataset for this system's additional features precaution and severity for the diseases will also be used. The architecture, data flow, and user interface of the system are all carefully taken into account throughout the designing stage. The objective is to produce a design that is simple to use and encourages interaction between users and the system. The elements and functions required to enable precise disease prediction are determined after extensive planning and investigation. The implementation phase starts when the design phase is finished. During this stage, the system must be developed in accordance with the predetermined criteria and the design specifications must be translated into actual code.

24 3.4.1 Data Collection The selected datasets include information on diseases, symptoms, precautions for symptoms, and severity which provides information about diseases, their associated symptoms, severity of the diseases, precaution for the diseases and descriptions of the diseases. Here is a detailed explanation of the datasets: 1) Diseases: The datasets contain a list of diseases and symptoms. Each disease entry includes the following attributes: a) Disease Name: The name of the disease. b) Symptoms: A list of symptoms associated with the disease. 2) Disease description: The datasets also provide a list of diseases and disease description. Each diseases entry includes the following attributes: a) Disease Name: The name of the disease. b) Overview of diseases: overview or explanation of diseases. 3) Symptom severity: The datasets provide a list of symptoms and the weightage of the severity for each symptom. Each symptom entry includes the following attributes: a) Symptom Name: The name of the symptom. b) Severity Weightage: The weightage of the severity from 1 to 7 which is low risk to high risk. 4) Disease precaution: The datasets provide a list of diseases and the potential precaution steps for each disease. Each disease entry includes the following attributes: a) Disease Name: The name of the disease. b) Precaution step: The potential step for each disease.

25 The datasets are structured in a way that allows for understanding the relationships between diseases and symptoms. Diseases are linked to their associated symptoms through the "Symptoms" attribute, which provides a list of symptom names associated with each disease. The data consist of 4920 sets of data about diseases and also symptoms. 2460 from the data are split to use for testing the machine learning. Figure 3.3 show example of the data. Figure 3.3 Example from the dataset

26 3.4.2 System Design Figure 3.3 shows the general overview of the system. When the users use the system, the user will be prompted to provide general medical info such as their height, weight, age and gender. After submitting the medical info, the users will prompt to provide the symptoms. The symptoms provided will be used as testing data for machine learning models to check on the accuracy. Based on the accuracy provided the system will retrieve top 5 possible diseases from the databases. Alongside that the system will retrieve the data about the diseases such as precaution and severity of each possible disease that have been diagnosed. All the data that have been retrieved will be displayed as output for the user as a diagnosis summary from the system. Figure 3.4 System architecture

27 3.4.3 Machine Learning Design Effective machine learning systems are created through a process called machine learning design. It includes all of the phases of strategic planning, modeling, and execution necessary to develop an effective machine learning solution. In this procedure, several aspects like data collection, preprocessing, feature engineering, algorithm selection, model training, and evaluation must be carefully taken into account. In addition, addressing issues with bias, fairness, interpretability, and scalability is a part of machine learning design. To build models that can accurately generalize and make trustworthy predictions on fresh, unforeseen data, a well-designed machine learning system takes into account the specific issue domain, the available data, and the desired results. The implementation chapter's primary focus is on the useful application of the Random Forest algorithm in a disease prediction system. This chapter covers in detail the Random Forest method's implementation approach for disease prediction, along with the issues and challenges specific to this subject. Figure 3.5 displays the visualization of the Random Forest method.

28 Figure 3.5 Random Forest Algorithm Illustration (Source: Sharma,2020) 1. Data collection: Gather a comprehensive dataset that includes relevant features or attributes related to the diseases being predicted. This dataset should have labeled data, where each instance is associated with the corresponding disease diagnosis. 2. Data preprocessing: Clean the dataset by handling missing values, removing irrelevant features, and normalizing or standardizing the data to ensure consistency and improve the performance of the algorithm. 3. Splitting the dataset: Divide the dataset into training and testing sets. The training set will be used to train the random forest model, while the testing set will be used to evaluate the model's performance. 4. Feature selection: Optionally, you can perform feature selection to identify the most informative features for disease prediction. This can help reduce the dimensionality of the dataset and improve the efficiency of the algorithm.

29 5. Training the random forest model: Use the training set to train the random forest algorithm. Random forest builds multiple decision trees using different subsets of the training data and random feature selection. Each tree is trained on a different subset of the data, and the final prediction is made by aggregating the predictions from all the trees. 6. Hyperparameter tuning: Tune the hyperparameters of the random forest algorithm, such as the number of trees, maximum tree depth, and minimum number of samples required to split a node, to optimize the model's performance. This can be done using techniques like cross-validation or grid search. 7. Evaluating the model: Once the model is trained, evaluate its performance on the testing set
using appropriate evaluation

62%	MATCHING BLOCK 9/27	SA	CSP650-FullReport_MuhammadHafizBinMohd.Saufan ... (D184813755)

metrics such as accuracy, precision, recall, and F1-score. This step helps assess how well the

random forest model predicts diseases. 8. Deployment: Once the model is deemed satisfactory, it can be deployed in the disease prediction system to make predictions on new, unseen data. This could involve integrating the model into a web application, mobile app, or any other suitable platform for practical use.

30 3.4.4 System UI Design Figure 3.7 displays the system's user interface, where users may either choose from a list of symptoms below or quickly search for symptoms using the search bar. By selecting the remove button located at the end of the selected symptom, the user can delete the chosen symptom. The user can begin diagnosing by pressing the diagnosing button after inputting symptoms. Figure 3.7: Input System UI

31 The outcome of the diagnosis is shown in Figure 3.8. The user can see a list of all potential diseases they may have on the left. The user can view the accuracy for every disease that has a chance of occurring. The user can follow the illnesses precaution step on the right side of the UI, which by default will show the disease description of the diseases at the top of the list. Figure 3.8 Output System UI
32 3.4.5 System Implementation The third phase of the waterfall paradigm is the implementation phase, which is also known as development. The implementation phase is typically the longest of the phases in the waterfall technique. The Python programming language will be used to implement the system in this stage. To ensure progressive improvement, the initial emphasis is on building a fundamentally sound, functional system that will be improved upon and improved upon through iterative testing cycles. 3.4.5.1 Hardware Requirement Hardware requirements for a disease prediction system can vary depending on the complexity and scale of the application. For this particular project, a processor with at least 4GB of RAM and an operating system of Windows 10 or higher are minimal hardware requirements. Any device capable of running web applications, such as laptops and desktops, is encouraged for system access and usage. Table 3.1 shows a general overview of the hardware requirements for such a system. Table 3.1 Hardware Specification Device Specification Laptop Model: Acer Processor: AMD Ryzen 3 3200U RAM: 8GB OS: Windows 11 Storage: 237GB
33 3.4.5.2 Software_Requirement Developing a disease prediction system requires specific software requirements to ensure its successful implementation. Firstly, an Integrated Development Environment (IDE) such as Visual Studio Code or PyCharm provides a suitable environment for coding, debugging, and testing. The choice of programming language depends on the application's requirements, but Python is commonly used for its extensive machine learning libraries. Web frameworks like Django or Flask facilitate web application development, while HTML, CSS, and JavaScript are essential for frontend design. Machine learning libraries like scikit-learn or TensorFlow enable the implementation and training of predictive models. Table 3.2 shows the overview of software requirements. Table 3.2 Software Requirement Integrated Development Environment (IDE) Visual Studio Code Programming Language Python Web Framework Flask Machine Learning Libraries TensorFlow , PyTorch Web Development Technologies HTML,CSS
34 3.5 Objective 3 Testing within the framework of the illness forecasting system. The testing phase of the system development lifecycle, which is essential for assuring the dependability, accuracy, and robustness of the illness prediction model, is thoroughly explored and discussed in this chapter. The chapter starts out by describing the goals of testing, which include confirming the system's functionality, assessing its performance, and locating potential areas for improvement. It highlights the value of thorough testing to find and fix any flaws or problems before deploying the system.
35 3.5.1 System Testing A thorough set of test cases has been constructed in order to validate the precision and usefulness of the disease prediction system. These test cases are made to evaluate the system's aptitude for dealing with diverse situations and potential problems. If any mistakes or weaknesses are found during testing, the development team will identify and fix them right away to ensure the program's overall quality. The test cases that will be used to extensively assess the performance of the illness prediction system are shown in Table 3.3. Table 3.3 User Acceptance Form NO Acceptance Requirement Critical Test Result Comment/Issue Yes No Accept Reject 1 Able to input data 2 Search features functional 3 Easy to use 4 Output easy to understand 5 Easy to navigate 6 All button functioning well 7 Able to produce expected result
36 3.6 Summary This chapter presents a thorough review of the project's research methods. The five phases of the research approach include requirement collecting and analysis, specification and identification, conversation design, architecture, and testing. A detailed literature research informs the beginning phase's task of determining the project's scope, objectives, and importance. The second stage then concentrates on determining the resources and software components needed to construct the prototype. Making a proposed design, which includes various visual representations like flowcharts and UI designs, is what the third phase requires. This phase tries to give a clear structure and visual representation of the system. The emphasis in the fourth phase is on setting the system apart from its competitors. This entails giving it special features or capabilities that make it stand out from other systems on the market. The prototype is run in the fifth and final phase to gauge performance and guarantee adequate functionality. Its operational effectiveness is rigorously tested in order to confirm it and find any potential weak points. The next chapter will provide insight on designing and development process which focusing on the implementation on this project.
37 CHAPTER 4 DESIGN AND DEVELOPMENT This chapter will conduct a thorough investigation of the procedures and experiments conducted during the study, providing insight into the results and deductions drawn from the use of diverse methodologies. This chapter carefully presents and examines the findings at every phase, which includes data collection, data pre-processing, feature extraction, model construction, model training, assessment, web creation, and a summary of the chapter as a whole. 4.1 DATA PRE-PROCESSING A methodical strategy was taken throughout the preprocessing stage to improve and polish the dataset's quality. A clean and trustworthy dataset for further analysis was ensured by methodically removing null data, which are known to create noise and inconsistencies. This first stage of the investigation not only improved the dataset's overall quality but also laid the groundwork for later phases to yield more precise and significant findings. Figure 4.1 show the process to identify null value in jupyter notebook. Figure 4.1 Code snippet null checker

38 A thorough attempt was made to standardize the representation of symptoms once null data was eliminated. This entailed establishing consistent nomenclature across the dataset by substituting underscores for spaces in the names of symptoms. This preprocessing phase was crucial in creating a consistent and organized basis for the future analyses by addressing differences in symptom naming practices, which added to the dataset's overall coherence. Figure 4.2 shows the result before removing null value. Figure 4.2 Visualization of dataset before preprocessing Additionally, formatting and structural flaws were also addressed as part of the dataset's improvement. A possible cause of disparities, trailing spaces, was methodically eliminated from the symptom columns. This reduced the possibility of errors resulting from inconsistent formatting in addition to improving the dataset's visual cleanliness. A more simplified and well-organized dataset that was ready for efficient analysis and modeling was the end result. Figure 4.3 shows the coding implemented for the preprocessing process.

39 Figure 4.3 Data preprocessing process A particular preprocessing step was applied in the context of symptom severity to deal with symptoms that had no assigned rank. A consistent representation was ensured by consistently assigning zero values to these symptoms. This method offered a workable option to deal with the lack of severity rankings in addition to adding to the dataset's numerical coherence. Every preprocessing step from removing null data to handling unranked symptoms contributed to the creation of a solid and polished dataset that was ready for the study's further phases. Figure 4.4 show the result of removing null value, removing trailing space, replace spaces with underscore for symptoms name and assign symptoms with no rank to zero value.

40 Figure 4.4 Result of dataset.csv after preprocessing Having now carefully preprocessed the dataset to guarantee its consistency and dependability, the research smoothly moves into Chapter 4.2, where the emphasis is placed on putting a Random Forest model into practice. With an eye on maximizing the Random Forest algorithm's predictive power for the given task, the following section expands on the revised dataset by exploring the nuances of model development, training, and assessment.

41 4.2 RANDOM FOREST MODEL The Random Forest model, a potent ensemble learning method renowned for its adaptability and solid performance, will be covered in detail in Chapter 4.2 along with the crucial stages of implementation. There are three main stages to this part, and each one is crucial to the entire process of developing a model. 4.2.1 DICTIONARY SETUP This section will provide insight in the process of setup and encoding the dataset for symptoms and diseases by defining the dictionary to map diseases with the symptoms and the symptoms with the severity weightage. Figure 4.5 shows the snippet for the encoding and mapping process. This snippet of code carries out several tasks associated with encoding the severity of symptoms in a given dataset. First, by building a dictionary (symptom_id_dict) that maps each symptom to its matching ID, it gives each symptom a distinct numerical ID. The 'severity' data frame has a new column called 'Symptom_ID' with this ID. Next, the assigned ID is multiplied by the relevant weight for each symptom to determine its severity. Next, two dictionaries are generated: symptom_id_dict, which maps symptoms to their IDs, and symptom_severity_dict, which maps symptoms to their weights. The severity columns in the 'data_severity' data frame are encoded with their appropriate weights, and the symptom columns in the dataset are encoded with the given symptom IDs. The values in 'data_severity' that are missing are represented by zeros. The result is saved in a new column called "totalseverity" in "data_severity." The total severity for each row is determined by adding together the severity values for each symptom. The 'severity' column in the main dataset is then assigned this total severity. Lastly, the first ten rows of the updated dataset are shown, and all missing values in the original dataset are replaced with zeros. In conclusion, this code creates a foundation for additional analysis and modeling by systematically capturing symptoms and their severity in the dataset.

42 Figure 4.5 Snippet for encoding and mapping process 4.2.2 FEATURE EXTRACTION The process starts with feature extraction, a critical stage in which the qualities of the dataset are carefully analyzed and chosen to create a complete set of features. In order to produce a structured representation that improves the model's ability to identify patterns and relationships within the data, this method entails extracting pertinent information from the preprocessed dataset. Figure 4.6 shows the process of feature extraction for this model in jupyter notebook. To put it briefly, the code divides the dataset into two sections: ‘labels’, which include the labels for the respective diseases, and ‘data’, which contains the characteristics (symptoms). Figure 4.6 Process of feature extraction.

43 4.2.3 BUILD MODEL After the features are set up, attention turns to building the Random Forest model. This include creating the framework that enables the model to generate well-informed predictions, as well as designing the ensemble's architecture and parameters. At this point, the Random Forest algorithm's subtleties become apparent, laying the groundwork for later training. Figure 4.7 shows the process of building the Random Forest model. This Python code uses the RandomForestClassifier from the scikit-learn module to define and train a Random Forest model. Firstly, the Random Forest technique for classification tasks is implemented by the RandomForestClassifier class, which is a part of the scikit-learn library. An ensemble learning technique called Random Forest builds a large number of decision trees during training and outputs the mean prediction (regression) or the mode of the classes (classification) of the individual trees. To delve deeper for further understanding this section will explain the parameter used. The option 'n_estimators=50' indicates the quantity of decision trees within the forest. There will be fifty trees in the forest in this instance. While more trees can result in improved speed, there is a trade-off between computational cost and performance. The parameter 'max_depth=10' regulates the maximum depth of every decision tree within the forest. Restricting the depth aids in avoiding overfitting. A tree with a depth of 10 can have a maximum of 10 decision node levels. The next parameter, 'min_samples_split=5', determines the bare minimum of samples needed to split an internal node. During tree development, it aids in regulating the size of partitions at each node. The parameter 'min_samples_leaf=2' is used to specify the minimum number of samples that must be present at a leaf node. It aids in regulating the decision trees' leaf sizes. The 'random_state=42' is the next option that sets the random number generator's beginning value. Reproducibility is ensured by planting a seed. Reproducibility in machine learning depends on using the same seed, which ensures to receive the same collection of random decisions during training.

44 Figure 4.7 Defining process of random forest model 4.2.4 MODEL TRAINING One of the most important stages is model training, where the algorithm learns from the labeled dataset to improve its predictive power. By exposing the model to the information, the training process makes it capable of identifying complex relationships, patterns, and correlations between the attributes. The model's comprehension is improved by this iterative process, guaranteeing its flexibility and accuracy in forecasting. Figure 4.8 shows the snippet code for splitting into 80% for training and 20% for testing. Figure 4.8 Process for splitting data into training and testing. This code is essential for evaluating the trained model's performance on the training set, providing information about possible overfitting or underfitting, and directing future model modifications as needed. Figure 4.9 shows the snippet code for model training using dataset that have been split. This code snippet evaluates the performance of a trained Random Forest Classifier model on the training data. Within this section, the formula 'y_pred = modelRFC.predict(x_train)' instructs the model to use the trained Random Forest Classifier (modelRFC) to predict (y_pred) on the input features (x_train). whereas the formula for calculating the model's accuracy is 'accuracy = accuracy_score(y_train, y_pred)'. This formula

100%	MATCHING BLOCK 10/27	SA	CSP650-FullReport_MuhammadHafizBinMohd.Saufan ... (D184813755)

compares the predicted labels (y_pred) with the actual labels (y_

train). This computation makes use of the scikit-learn accuracy_score function.

45 Figure 4.9 Code Snippet for model training As we transition to Chapter 4.3, which delves into the world of web development and integration, the results of our Random Forest Model provide the framework for a user- friendly interface that makes the predictive power easily accessible and useful to a larger user base. The model's successful implementation in Chapter 4.2 opens the door to a more thorough investigation of real-world applications during the subsequent web development stage. 4.2.5 MODEL TESTING A critical step in assessing the effectiveness and dependability of the created illness prediction system is model testing. This section explores the used assessment metrics, emphasizing the Confusion Matrix in particular. Figure 4.10 show the visualization for confusion matrix. Figure 4.10 Confusion Matrix

46 4.3 WEB BUILDING This section focuses on the critical process of using web development to transform the machine learning-based illness diagnosis algorithm into a useful and approachable interface. The incorporation of machine learning into web apps has been crucial in delivering effective and user-friendly solutions in the age of technological breakthroughs. Web Integration (4.3.1) and Web Development (4.3.2) are the two main components involved in developing the web interface for the Diseases Diagnosis System. 4.3.1 WEB INTERGRATION Web integration involves the smooth integration of the web-based platform with the machine learning model for illness detection. The frontend user interface and the backend machine learning algorithms work together seamlessly thanks to this connection. A strong link between these components allows users to enter symptoms with ease and get precise predictions, creating a dynamic and engaging user experience. The 'joblib' is used in the context of web integration, machine learning models are usually handled effectively. The trained machine learning model is loaded and saved using the 'joblib' function in the provided code snippet on Figure 4.11. 'trained_model.joblib' is the file where the Random Forest Classifier (modelRFC) is saved after it has been trained using the joblib.dump() function. In particular, in a web application where the model needs to be loaded fast, this step is essential for maintaining the trained model's state so that it can be reused without retraining.

47 Figure 4.11 Code snippet on the use of joblib To reload the saved model into the application, use the joblib.load() function. This saves time and computational resources by enabling instant access to the pre-trained machine learning model without the need to retrain it. Figure 4.12 shows the used of joblib.load() function in app.py flask file. This code is used to load data from modelRFC using joblib.load('model_and_data.joblib'). The loaded object then assigned to each of the new variable. The notable object loaded are modelRFC which is the model that has been train before, the dataset that have been preprocess, symptom_severity_dict for the symptoms and severity weight and symptom_id_dict for the symptom that have been mapped with its symptom id. Figure 4.12 Code snippet on utilizing joblib library. Using joblib to save and load machine learning models guarantees that the trained model is always available for use in web applications for prediction purposes. When implementing machine learning models in production contexts, this method is very helpful since it allows for quick and efficient model access, which is necessary to

48 provide users interacting with the web interface with real-time predictions. The building of a user-friendly and effective web interface is a crucial step that transition from the application of machine learning models into Chapter 4.3.2, which describes the complex process of web development for the diseases prediction system. 4.3.2 WEB DEVELOPMENT The design and development of the user interface, with a focus on accessibility and user-friendliness, is included in the Web Development process. This part entails putting into practice an aesthetically pleasing and user-friendly design that makes it simple for users to enter symptoms and understand diagnosis outcomes. During the development process, other factors that need to be taken into account are the addition of features that improve user understanding on the results shown. 4.3.2.1 DIAGNOSIS PAGE Figure 4.13 show the code snippet of Flask from app.py. The @app.route('/diagnosis') decorator in the given Flask code specifies the route for the root URL ("/diagnosis"), and the related home() method manages GET and POST requests. The 'index.html' template is rendered when a user navigates to the root URL, which initiates a GET request. The render_template function indicates that this template is in charge of showing the user interface. Figure 4.13 Flask code snippet for diagnosis page The code excerpt from index.html for the user-selectable symptom option is displayed in Figure 4.14. The user is presented with the possible symptom alternatives within the 'index.html' for choosing. The symptom_options variable, which is provided to the index.html from the Flask application, enables this interaction. A list of symptom

49 names gleaned from the symptom severity dataset is contained in this variable. Because of this integration's dynamic nature, the web interface may directly pull symptoms from the dataset, providing users with an extensive and current list of symptoms to choose from. Figure 4.14 Code snippet from index html. The web development approach guarantees a smooth connection between the frontend (HTML template) and backend (Flask application) by utilizing Flask's template rendering and routing mechanism. This allows users to choose symptoms in the disease diagnosis system in an interactive and user-friendly manner.

50 4.3.2.2 RESULT PAGE For the result.html where the output of the system took place, Flask, a Python web framework, is used in the web development part of the disease prediction system based on symptoms utilizing machine learning to provide smooth interactions between the user interface and the backend. Javascript queries are handled by two crucial routes, /get_description and /get_precautions, which dynamically retrieve relevant data from designated databases. Figure 4.15 show the code snippet for route /get_description. Requests for illness descriptions are answered via the /get_description route. The illness name is extracted from the request parameters by the related get_description() function upon receipt of a GET request. The code then retrieves the description associated with the given ailment by utilizing a DataFrame called symptom_description_dataset. After that, jsonify({'success': True, 'description': description}) is used to bundle this description as a JSON response, enabling dynamic integration into the'result.html' template. Figure 4.15 Code snippet for /get_description route Figure 4.16 show the code snippet for route /get_precaution route. Similar to before, requests for disease precautions are handled via the /get_precautions route. The get_precautions() function is called in response to a GET request. It first retrieves the name of the disease and then, if a DataFrame called symptom_precaution_dataset with disease precautions is present, retrieves the precautions associated with the given disease. Using jsonify({'success': True, 'precautions': precautions.tolist()}), these precautions are converted into a list and contained within a JSON response. This method guarantees that, based on user-selected diseases, the'result.html' template can dynamically incorporate current information about precautions as well as disease descriptions, improving the user experience overall.

51 Figure 4.16 Code snippet for /get_precautions route 4.3.2.3 JAVASCRIPTS The critical JavaScript implementation is covered in detail in Chapter 4.3.2.1 in the context of the web development project for disease diagnostics. JavaScript is essential for improving the user interface's responsiveness and engagement because it offers dynamic capability for selecting symptoms, searching, and starting the diagnosis procedure. The complexities of the JavaScript functions used in the web application will be covered in detail in this section. These functions will explain how they handle user input, update the symptom options in real-time, manage the list of selected symptoms, and trigger the diagnosis with a smooth integration into the Flask framework. This chapter attempts to provide light on the complex web development procedures that go into creating a more interesting and user-friendly illness detection system by exploring the subtleties of JavaScript implementation. The JavaScript code that uses the Flask framework to integrate a symptoms selection system into a web interface is displayed in this section. A button click triggers the removeAllSymptoms function, which removes the chosen symptoms list. formatSymptom capitalizes the first letter and inserts spaces in place of underscores to guarantee consistency in formatting. Figure 4.17 shows that based on user input, the addSelectedSymptom function dynamically modifies the available symptom options. It handles symptom selection via the updateSymptom function, updating the list of selected symptoms as necessary. Real-time search capabilities was provided by the searchInp event listener, which filters symptoms based on user input.

52 Figure 4.17 Code snippet for addSelectedSymptom() function Figure 4.18 show the diagnosis procedure is started by the startDiagnosis function after the user chooses three symptoms or more. It updates the page with the diagnosis result after retrieving the data from the server. The function showDiseaseDescription retrieves and shows details about a disease, such as its description, any necessary precautions, and the severity level. The code demonstrates asynchronous actions, which enhance user experience by updating data smoothly and avoiding the need to reload the entire page. Figure 4.18 Code snippet for startDiagnosis() function

53 The use of JavaScript to improve the functionality of the web-based illness diagnosis system is covered in detail in Chapter 4.3.2.1. This section highlights how important JavaScript is to provide dynamic interactions, symptom selection, and intuitive updates in the web interface. The foundation of smooth integration and instantaneous response, which enhances user experience, is provided by JavaScript. This synergy is further extended in the following Chapter 4.4, which delves into the complexities of web design to improve the overall look and feel of the illness diagnosis platform. The joint usage of web design and JavaScript ensures a unified and captivating user experience, enhancing the system's effectiveness and usability. 4.4 SUMMARY In summary, Chapter 4 presents a thorough implementation of a system for diagnosing diseases based on symptoms that smoothly combines web development and machine learning. Thorough pre-processing takes care of all the details of data formatting and cleansing, which sets the groundwork for a strong model. Using a Random Forest model for feature extraction, model construction, and training creates a powerful foundation for precise illness predictions. Web development extends the functionality by smoothly integrating JavaScript and Flask and offers a user-friendly symptom selection interface. The web application provides comprehensive disease information by utilizing dynamic updates and integrating with symptom severity datasets. This combination of cutting-edge algorithms and intuitive user interfaces shows promise for revolutionizing symptom-based disease detection and might have a big impact on healthcare usability and accessibility.

54 CHAPTER 5 RESULT AND DISCUSSION This crucial chapter will examine the results of the illness prediction system and assess its functionality and performance. Three primary sections comprise the full examination: Functionality Testing (5.1), Summary of the Test Cases (5.2), and Conclusion (5.3). 5.1 Functionality Testing Since functionality testing is a prerequisite for each waterfall stage, it is a crucial component of every project. This testing stage ensures that a software program complies with design guidelines, runs without a hitch, and doesn't include any errors when users interact with it. It evaluates the acceptability of the system's operation as well. Testing of functionality is displayed in Table 5.1. Table 5.1 Functionality test Test Case Expected Result Success/Failure View homepage The system will show the homepage. Start button The system will start the main features. Search symptom The search bar capable of finding and filter symptom based on user input. Select symptom The symptom option can be select as the option for diagnosis. Remove symptom Remove button can remove the selected

55 symptom the user does not want to include from selected symptom field Remove all symptom Remove all button will remove all the symptom from the selected symptom field Start diagnosis Button start diagnosis will redirect user to the result page where the system shows the result of the diagnosis based on the selected symptom. Diseases' card Clicking on the diseases card will show the diseases description and the precaution steps for the diseases. Start new diagnosis Button start diagnosis will redirect user to the diagnosis page where user can start a new diagnosis. Alert Message Show alert message when start diagnosis with insufficient symptoms

56 5.1.1 Test Case 1: View Homepage Table 5.2 Test Case 1 Description User landing page. Test Summary To confirm the page will redirect to homepage. Related Page Homepage. Potential Output Homepage as landing page. Activities Start the system. Expected Output The homepage will be displayed. Actual Output Figure 5.1 Homepage Status Pass

57 5.1.2 Test Case 2: Start Button Table 5.3 Test Case 2 Description Start the main features Test Summary To confirm the page will redirect to redirect to diagnosis page. Related Page Homepage and Diagnosis page. Potential Output Diagnosis page. Activities Click the start button. Expected Output The diagnosis page will be displayed. Actual Output Figure 5.2 Diagnosis page Status Pass

58 5.1.3 Test Case 3: Search Symptoms Table 5.4 Test Case 3 Description Find and filter symptoms based on user input. Test Summary To confirm the search button can find and filter the symptoms based on user input. Related Page Diagnosis page. Potential Output Filtered symptom based on search input. Activities Type input on search box. Expected Output If user type in 'bl' the symptom option will show symptom with initial start with 'bl'. Actual Output Figure 5.3 Search Function Status Pass

59 5.1.4 Test Case 4: Select Symptom Table 5.5 Test Case 4 Description User select symptoms. Test Summary To confirm the symptoms selection can be choose. Related Page Diagnosis page. Potential Output Selected symptoms on the selected symptoms field. Activities Click on symptoms that user want to choose. Expected Output When user click on the symptoms from the selection it will show in the selected symptom on the right of the site. Actual Output Figure 5.4 User selection Status Pass

60 5.1.5 Test Case 5: Remove Symptom Table 5.6 Test Case 5 Descriptio n Remove the selected symptom from selected symptom field. Test Summary To confirm the remove button will remove the selected symptom using the remove button. Related Page Diagnosis page. Potential Output Selected symptom will be removed. Activities Click on the remove button to remove selected symptom. Expected Output The abdominal pain will be removed when click on the remove button on the symptom right side. Actual Output Figure 5.5 Remove function Status Pass

61 5.1.6 Test Case 6: Remove All Symptoms Table 5.7 Test Case 6 Descriptio n Remove all the symptoms in the selected symptoms field. Test Summary To confirm the remove all button capable to remove all the selected symptoms. Related Page Diagnosis page. Potential Output No symptom in selected symptoms field. Activities Click on the remove all button. Expected Output The selected symptoms field will be blank. Actual Output Figure 5.6 Remove All Function Status Pass

62 5.1.7 Test Case 7: Start Diagnosis Table 5.8 Test Case 7 Description Start the diagnosis process. Test Summary To confirm the start diagnosis button will redirect user to the result page for the predicted diseases results. Related Page Diagnosis page and result page. Potential Output Result page with prediction results. Activities Click on the start diagnosis button. Expected Output The user will redirect to result page if the input of the selected symptom is sufficient. Actual Output Figure 5.7 Result page Status Pass

63 5.1.8 Test Case 8: Diseases Descriptions and Precaution Table 5.9 Test Case 8 Description Show the diseases description and precaution. Test Summary To confirm that the diseases card will shows the diseases description and precaution when clicked. Related Page Result page. Potential Output Diseases and precaution with its severity will be shown. Activities Click any of the predicted diseases on the result section. Expected Output Disease description and the precaution will be shown on the right section of the website. Actual Output Figure 5.8 Diseases and Precaution Result Status Pass

64 5.1.9 Test Case 9: Start New Diagnosis Table 5.10 Test Case 9 Description Start a new diagnosis process. Test Summary To confirm the start new diagnosis button will redirect user to the diagnosis page for the diagnosis process. Related Page Result page and diagnosis page. Potential Output Diagnosis page. Activities Click on the start new diagnosis button. Expected Output The user will redirect to diagnosis page. Actual Output Figure 5.9 Diagnosis page Status Pass

65 5.1.10 Test Case 10: Alert Message Table 5.11 Test Case 10 Description Alert message will pop up when the selected symptoms is insufficient. Test Summary To confirm alert message will pop up when the user start diagnosis with insufficient symptoms. Related Page Diagnosis page. Potential Output Alert message will pop up. Activities Click start diagnosis with none selected symptom on selected symptoms field. Expected Output The page will not redirect user to result page instead show the user an alert message. Actual Output Figure 5.10 Alert Message Status Pass

66 5.2 Summary of Test Cases Table 5.12 Summary of Test Cases No Test Case Success/Failure 1 View homepage Success 2 Start button Success 3 Search symptom Success 4 Select symptom Success 5 Remove symptom Success 6 Remove all symptom Success 7 Start diagnosis Success 8 Disease description Success 9 Start new diagnosis Success 10 Alert Message Success 5.3 Conclusion The system's overall performance in these test cases confirms its dependability, ease of use, and capacity to fulfill its primary goal of providing accurate disease prediction based on user-provided symptoms. In conclusion, all of the system's components have passed functional testing, proving that all of the features are working as intended. The purpose of these tests was to make sure the system was operating correctly. Moving on to Chapter 6, "Conclusion," the end of the thorough testing procedure lays the groundwork for a thorough comprehension of the system's capabilities and prepares the reader for contemplative thoughts and further considerations.

67 CHAPTER 6 CONCLUSION AND RECOMMENDATION This chapter provides a thorough analysis of the project's goals, constraints, and suggested next steps in the endeavor to develop a novel and effective illness prediction system. The usefulness of these elements as a whole and future developments in the field of health informatics are heavily dependent on their performance. 6.1 Objective This project's main goal was to create a disease prediction system that uses web-based technologies and machine learning methods to be dependable and accurate. As the project examine the particular goals, its accomplishments highlight how strong the solutions that have been put in place are. This section will conclude overall objective that have been achieve throughout the project. 6.1.1 Objective 1 Finding appropriate methods and algorithms for identifying possible diseases based on input symptoms was the goal of Objective 1. The accomplishment of this goal is evidence of the careful selection and application of cutting-edge illness prediction techniques. Numerous studies have demonstrated that the random forest algorithm is a particularly good method for diagnosing diseases because it can manage intricate relationships, accommodate a wide range of symptoms, and reduce overfitting issues. All of these factors ultimately improve the precision and dependability of the disease prediction system.

68 6.1.2 Objective 2 Goal 2 was to create an intuitive web-based system that adhered to the selected methods and algorithms. The accomplishment of this goal emphasizes how well technology is integrated to provide a user-friendly platform. To sum up, Flask is an excellent framework for integrating the selected methods and algorithms into a user- friendly web-based system for disease diagnosis because of its lightweight design, flexibility, integration with the Python ecosystem, built-in development server, and strong community support. 6.1.3 Objective 3 Thorough testing of the system's functionality and confidence level was the third objective. The positive results in this area confirm the system's capacity to generate accurate disease forecasts and the resilience of the applied algorithms. The system's capacity to give accurate disease predictions is validated by the successful testing process, which highlights the resilience of the applied algorithms. As a result, all test cases were successfully completed, demonstrating the system's dependability, functionality, and user-friendliness. This also validates the resilience of the applied algorithms and the system's capacity to produce reliable and accurate illness forecasts. 6.2 Limitation Despite the accomplishments, it is imperative to recognize the systemic constraints. One significant drawback is the data's accessibility, which could affect the results' degree of confidence. This constraint forces a more thoughtful assessment of the system's bounds and paves the way for upcoming enhancements. One significant drawback that needs to be carefully considered is the lack of diversity and amount in the dataset used to train and evaluate the machine learning models. This limitation is a crucial factor that affects the dependability and degree of confidence in the system's outputs, not just a barrier. The richness and representativeness of the data the system processes have a direct impact on how well it predicts diseases. Thinking back on this

69 restriction, it acts as a spur for upcoming projects that will be covered in more detail in the part that follows. 6.3 Future Recommendation This project's future course should be carefully planned with a greater emphasis on machine learning research. The ongoing investigation of novel approaches is necessary in the search for the best algorithms to forecast illnesses. It is recommended that scholars and professionals explore the broad field of machine learning, test various models, and evaluate their effectiveness in the context of prognosticating diseases. Given how important data is to the performance of machine learning models, efforts going forward should focus on growing the system's data store. Crucial tactics include working with healthcare organizations, utilizing real-time data sources, and making sure that different demographic groups are represented. In addition to addressing the present shortcoming, the data augmentation moves the system closer to a more thorough and precise illness prediction. Investing in UI research and design can enable users to easily traverse the system, including self-diagnosing individuals and healthcare professionals. In addition to improving user happiness, a UI that is aesthetically pleasing, responsive, and easy to use also makes the system more usable and efficient. In conclusion, the recommendations for the future are based on a dual commitment that prioritizes user-centric advancements for a more engaging and approachable experience while also pushing machine learning advances to advance the technological core.

70 REFERENCES Abdar, M., Zomorodi-Moghadam, M., Zhou, X., Gururajan, R., Tao, X., Barua, P. D., & Gururajan, R. (2020). A new nested ensemble technique for automated diagnosis of breast cancer. Pattern Recognition Letters, 132, 123–131.
https://doi.org/10.1016/j.patrec.2018.11.004 Acharya, B., & Sahu, K. (2020). Software Development Life Cycle Models: A Review Paper. International Journal of Advanced Research in Engineering and Technology, 11(12), 169–176.
https://doi.org/10.34218/IJARET.11.12.2020.019 Aggarwal, K., Mijwil, M. M., Sonia, Al-Mistarehi, A. H., Alomari, S., Gök, M., Zein Alaabdin, A. M., & Abdulrhman, S. H. (2022).

100% **MATCHING BLOCK 11/27** **W**

Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning. Iraqi Journal for Computer Science and Mathematics, 3(1), 115–123.

https://doi.org/10.52866/ijcsm.2022.01.01.013 Ali*, N. shabaz, & Divya, G. (2020).

100% **MATCHING BLOCK 12/27** **W**

Prediction of Diseases in Smart Health Care System using Machine Learning.

International Journal of Recent Technology and Engineering (IJRTE), 8(5), 2534–2537. https://doi.org/10.35940/ijrte.E6482.018520 Al-Saqqa,

86% **MATCHING BLOCK 13/27** **W**

S., Sawalha, S., & Abdelnabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11), 246–270. https://doi.org/10.3991/ijim.v14

i11.13269 Angela Adanna, A., & Francisca Nonyelum, O. (2020). CRITERIA FOR CHOOSING THE RIGHT SOFTWARE DEVELOPMENT LIFE CYCLE METHOD FOR THE SUCCESS OF SOFTWARE PROJECT (Vol. 1). Aroral, H. K. (n.d.).

100% **MATCHING BLOCK 14/27** **SA** REPORT-2019294656.docx (D142598150)

Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis. International Journal of Applied Business and Management Studies, 6(1), 2021.

Banerjee, N. (2020). Prediction Lung Cancer-In Machine Learning Perspective. In 2020 International Conference on Computer Science, Engineering and Applications (ICCSEA).

100% **MATCHING BLOCK 15/27** **W**

Battineni, G., Sagaro, G. G., Chinatalapudi, N., & Amenta, F. (2020). Applications of machine learning predictive models in the chronic disease diagnosis.

In Journal of Personalized Medicine (Vol. 10, Issue 2). MDPI AG. https://doi.org/10.3390/jpm10020021 Black, C. J., Drossman, D. A., Talley, N. J., Ruddy, J., & Ford, A. C. (2020). Functional gastrointestinal disorders: advances in understanding and management. In The Lancet (Vol. 396, Issue 10263, pp. 1664–1674). Lancet Publishing Group. https://doi.org/10.1016/S0140-6736(20)32115-2 Chauhan, T., Rawat, S., Malik, S., & Singh, P. (2021). Supervised and Unsupervised Machine Learning based Review on Diabetes Care. 2021 7

100%	MATCHING BLOCK 16/27	W

th International Conference on Advanced Computing and Communication Systems, ICACCS 2021, 581–585.

https://doi.org/10.1109/ICACCS51430.2021.9442021 Chen, Y.-W., & Jain, L. C. (n.d.). Intelligent Systems Reference Library 171 Deep Learning in Healthcare Paradigms and Applications. http://www.springer.com/series/8578 Henrique, B. M., Sobreiro, V. A., & Kimura, H. (2019). Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications, 124, 226–251. https://doi.org/10.1016/j.eswa.2019.01.012
71 Ibrahim, I., & Abdulazeez, A. (2021).

100%	MATCHING BLOCK 17/27	W

The Role of Machine Learning Algorithms for Diagnosing Diseases. Journal of Applied Science and Technology Trends, 2(01), 10– 19.

https://doi.org/10.38094/jastt20179 Institute of Electrical and Electronics Engineers. (n.d.). 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS) : proceedings : Amman, Jordan, 9-11 October 2019. Islam, M. Z., Islam, M. M., & Asraf, A. (2020).

100%	MATCHING BLOCK 18/27	W

A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images.

Informatics in Medicine Unlocked, 20. https://doi.org/10.1016/j.imu.2020.100412 Islam Trishna, T., Imran Hossen Sajal, G., Uddin Emon, S., Kundu, S., Rahman Ema, R., & Islam, T. (n.d.).

92%	MATCHING BLOCK 19/27	W

Detection of Hepatitis (A, B, C and E) Viruses Based on Random Forest, K-nearest and Naïve Bayes Classifier.

Jha, K., Doshi, A., Patel, P., & Shah, M. (2019). A comprehensive review on automation in agriculture using artificial intelligence. In Artificial Intelligence in Agriculture (Vol. 2, pp. 1–12). KeAi Communications Co. https://doi.org/10.1016/j.aiia.2019.05.004

86%	MATCHING BLOCK 20/27	W

Kamble, S. D., Patel, P., Fulzele, P., Bangde, Y., Musale, H., & Gaddamwar, S. (2021). Disease Diagnosis System using Machine Learning. Journal of Pharmaceutical Research International, 185–194. https://doi.org/10.9734/jpri/2021/v33i33b31810

Kapoor, A., Guha, S., Kanti Das, M., Goswami, K. C., & Yadav, R. (2020). Digital healthcare: The only solution for better healthcare during COVID-19 pandemic? In Indian Heart Journal (Vol. 72, Issue 2, pp. 61–64). Elsevier B.V.

59%	MATCHING BLOCK 21/27	W

https://doi.org/10.1016/j.ihj.2020.04.001 Kumar, N., Narayan Das, N., Gupta, D., Gupta, K., & Bindra, J. (2021). Efficient Automated Disease Diagnosis Using Machine Learning Models. Journal of Healthcare Engineering, 2021. https://doi.org/10.1155/2021/9983652

Lalband, N., & Kavitha, D. (2019). Software engineering for smart healthcare applications. International Journal of Innovative Technology and Exploring Engineering, 8(6 Special Issue 4), 325–331.

85%	MATCHING BLOCK 22/27	W

https://doi.org/10.35940/ijitee.F1066.0486S419 Magnavita, N., Tripepi, G., & Di Prinzio, R. R. (2020). Symptoms in health care workers during the covid-19 epidemic. A cross-sectional survey. International Journal of Environmental Research and Public Health, 17(14), 1–15.

https://doi.org/10.3390/ijerph17145218

91%	MATCHING BLOCK 23/27	W

Miller, S., Gilbert, S., Virani, V., & Wicks, P. (2020). Patients⇔ utilization and perception of an artificial intelligence⇓based symptom assessment and advice technology in a British primary care waiting room: Exploratory pilot study. JMIR Human Factors, 7(3). https://doi.

org/10.2196/19713 Mohan, J., Gupta, A., Jaypee Institute of Information Technology University, Institute of Electrical and Electronics Engineers. Uttar Pradesh Section. SP/CS Joint Chapter, & Institute of Electrical and Electronics Engineers. (n.d.). 2019 International Conference on Signal Processing and Communication (ICSC) : 07-09 March 2019, Jaypee Institute of Information Technology, NOIDA. Poongodi, M., Sharma, A., Vijayakumar, V., Bhardwaj, V., Sharma, A. P., Iqbal, R., & Kumar, R. (2020). Prediction of the price of Ethereum blockchain cryptocurrency in an industrial finance system. Computers and Electrical Engineering, 81. https://doi.org/10.1016/j.compeleceng.2019.106527
72 Quer, G., Arnaout, R., Henne, M., & Arnaout, R. (2021). Machine Learning and the Future of Cardiovascular Care: JACC State-of-the-Art Review. In Journal of the American College of Cardiology (Vol. 77, Issue 3, pp. 300–313). Elsevier Inc. https://doi.org/10.1016/j.jacc.2020.11.030 Ramirez, A., Aiello, A., & Lincke, S. J. (2020, November 26). A survey and comparison of secure software development standards. 13th CMI Conference on Cybersecurity and Privacy - Digital Transformation - Potentials and Challenges, CMI 2020. https://doi.org/10.1109/CMI51275.2020.9322704 Rodziewicz, T. L., & Hipskind, J. E. (n.d.). Medical Error Prevention. https://www.ncbi.nlm.nih.gov/books/NBK499956/ Rong, G., Mendez, A., Bou Assi, E., Zhao, B., & Sawan, M. (2020). Artificial Intelligence in Healthcare: Review and Prediction Case Studies. In Engineering (Vol. 6, Issue 3, pp. 291–301). Elsevier Ltd. https://doi.org/10.1016/j.eng.2019.08.015 Royce, C. S., Hayes, M. M., & Schwartzstein, R. M. (2019).

92%	MATCHING BLOCK 24/27	W

Teaching Critical Thinking: A Case for Instruction in Cognitive Biases to Reduce Diagnostic Errors and Improve Patient Safety. In Academic Medicine (

Vol. 94, Issue 2, pp. 187–194). Lippincott Williams and Wilkins. https://doi.org/10.1097/ACM.0000000000002518 Saravanan, T., Jha, S., Sabharwal, G., & Narayan, S. (2020). Comparative Analysis of Software Life Cycle Models. Proceedings - IEEE 2020 2nd

100%	MATCHING BLOCK 25/27	SA	FINAL REPORT_MUHAMMAD HANIF BIN ADNAN_20229128 ... (D184517383)

International Conference on Advances in Computing, Communication Control and Networking,

ICACCCN 2020, 906–909. https://doi.org/10.1109/ICACCCN51052.2020.9362931 Sri Eshwar College of Engineering, & Institute of Electrical and Electronics Engineers. (n.d.). 2020 6

100%	MATCHING BLOCK 26/27	W

th International Conference on Advanced Computing and Communication Systems (ICACCS).

Yu, H., Mohammed, F. O., Abdel Hamid, M., Yang, F., Kassim, Y. M., Mohamed, A. O., Maude, R. J., Ding, X. C., Owusu, E. D. A., Yerlikaya, S., Dittrich, S., & Jaeger, S. (2023).

82%	MATCHING BLOCK 27/27	W

Patient-level performance evaluation of a smartphone-based malaria diagnostic application. Malaria Journal, 22(1). https://doi.org/10.1186/s12936-023- 04446-0

Zhang, X., He, T., & Shen, W. (2020). Comparison of physical examination, ultrasound techniques and magnetic resonance imaging for the diagnosis of deep infiltrating endometriosis: A systematic review and meta-analysis of diagnostic accuracy studies. Experimental and Therapeutic Medicine. https://doi.org/10.3892/etm.2020.9043
73 APPENDICES
74 APPENDIX A: Gant Chart Research Study

Hit and source - focused comparison, Side by Side

Submitted text	As student entered the text in the submitted document.
Matching text	As the text appears in the source.

1/27	SUBMITTED TEXT 235 WORDS	53% MATCHING TEXT 235 WORDS
	BACHELOR OF COMPUTER SCIENCE (HONS.) February 2024 ii Universiti Teknologi MARA Diseases Prediction Based on Symptoms Using Machine Learning Technique Mohamad Afham Aiman Bin Mohamad Asri Thesis submitted in fulfillment of the requirements for Bachelor of Computer Science (Hons.) College of Computing, Informatics and Media February 2024 iii SUPERVISOR'S APPROVAL Diseases Prediction Based on Symptoms Using Machine Learning Technique By MOHAMAD AFHAM AIMAN BIN MOHAMAD ASRI 2022981099 This thesis was prepared under the supervision of project supervisor, Dr. Firdaus Fadzil. It was submitted to the College of Computing, Informatics and Media and was accepted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Hons) Computer Science. Approved by … … … … … … … … … … … Dr. Firdaus Fadzil Project Supervisor February, 1,2024 iv STUDENT'S DECLARATION I certify that this report and project to which it refers are the product of my own work and that any idea or quotation from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referring practices of the discipline. … … … … … … … … … …	BACHELOR OF COMPUTER SCIENCE(HONS.) FEBRUARY 2024 UNIVERSITI TEKNOLOGI MARA MALWARE DETECTION ON ANDROID USING RANDOM FOREST MUHAMMAD HAFIZ BIN MOHD. Thesis submitted in fulfillment of the requirements for Bachelor Computer Science (HONS.)Faculty of Computer and Mathematical Sciences FEBRUARY 2024 1 SUPERVISOR APPROVAL MALWARE DETECTION ON ANDROID USING RANDOM FOREST By MUHAMMAD HAFIZ BIN MOHD. This thesis was prepared under the supervision of the project supervisor, Miss Samsiah Binti Ahmad. It was submitted to the Faculty of Computer and Mathematical Sciences and was accepted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Hons.) Approved by Miss Samsiah Binti Ahmad Project Supervisor 2 STUDENT DECLARATION I certify that this thesis and the project to which it refers is the product of my own work and that any idea or quotation from the work of other people, published or otherwise are fully acknowledged in accordance with the standard referring practices of the discipline.

SA CSP650-FullReport_MuhammadHafizBinMohd.Saufan (1).pdf (D184813755)

2/27	SUBMITTED TEXT 57 WORDS	43% MATCHING TEXT 57 WORDS
	TABLE OF CONTENTS vi LIST OF TABLES ix LIST OF FIGURES x LIST OF ABBREVIATIONS xi CHAPTER 1: INTRODUCTION 1 1.1 Background of Study 2 1.2 Problem Statement 3 1.3 Project Aim 3 1.4 Project Objective 3 1.5 Project Scope 3 1.6 Project Significance 3 1.7 Project Outline 4 CHAPTER 2:	Table of Contents List of Tables 4 List of Figures 5 List of Abbreviations 6 CHAPTER ONE: PROJECT MOTIVATION 8 1.1 Background Study 8 1.2 Problem Objectives 10 1.4 Project Scope 10 1.5 Significance of The Project 10 1.6 Chapter

SA Proposal Report_Nur Alia Shakila_CS2305B.docx (D125615167)

3/27	SUBMITTED TEXT	26 WORDS	52% MATCHING TEXT	26 WORDS

it is difficult to identify a consistent source of errors and to offer a consistent workable solution that reduces the likelihood of a recurrent event (

It is challenging to uncover a consistent cause of errors and, even if found, to provide a consistent viable solution that minimizes the chances of a recurrent event.

W https://www.ncbi.nlm.nih.gov/books/NBK499956/

4/27	SUBMITTED TEXT	11 WORDS	100% MATCHING TEXT	11 WORDS

The aim of this project is to develop a system

SA rithikanew (4) (2).docx (D105034607)

5/27	SUBMITTED TEXT	17 WORDS	62% MATCHING TEXT	17 WORDS

The background of the study, problem statement, aim, objectives, significance, scopes and outline of the project

the background of the study, problem statement, objectives, scope, and significance of the project,

SA Amir Mustaqim Bin Aman_2022912795_CSP650 Final Report Submission.pdf (D184285955)

6/27	SUBMITTED TEXT	25 WORDS	58% MATCHING TEXT	25 WORDS

anosmia and dysgeusia in COVID-19 cases was reported to be 100.7 (95% Confidence Interval [CI] = 26.5-382.6) and OR = 51.8 (95%CI 16.6- 161.9), respectively,

Anosmia and dysgeusia in COVID-19 cases were found to have an odds ratio (OR) = 100.7 (95% Confidence Interval [CI] = 26.5–382.6) and an OR = 51.8 (95%CI 16.6–161.9), respectively.

W https://doi.org/10.3390/ijerph17145218

7/27	SUBMITTED TEXT	14 WORDS	100% MATCHING TEXT	14 WORDS

is a popular machine learning algorithm used for both classification and regression tasks,

SA report To check Pilgirism.pdf (D172727672)

8/27	SUBMITTED TEXT	16 WORDS	96% MATCHING TEXT	16 WORDS

waterfall model. Figure 3.1 Waterfall Model Illustration 20 (Source: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm)

Waterfall model illustration. Figure 2.8 Waterfall Model Illustration (Source: https://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm #)

SA Batrisyia Ajuwa binti Rozaini_Final Report.pdf (D158294162)

9/27	SUBMITTED TEXT	16 WORDS	62% MATCHING TEXT	16 WORDS
	metrics such as accuracy, precision, recall, and F1-score. This step helps assess how well the		metrics such as accuracy, precision, recall, and F1-score to evaluate how well the	

SA CSP650-FullReport_MuhammadHafizBinMohd.Saufan (1).pdf (D184813755)

10/27	SUBMITTED TEXT	10 WORDS	100% MATCHING TEXT	10 WORDS
	compares the predicted labels (y_pred) with the actual labels (y_		compares the predicted labels (y_pred) with the actual labels (y_	

SA CSP650-FullReport_MuhammadHafizBinMohd.Saufan (1).pdf (D184813755)

11/27	SUBMITTED TEXT	25 WORDS	100% MATCHING TEXT	25 WORDS
	Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning. Iraqi Journal for Computer Science and Mathematics, 3(1), 115–123.		Has the Future Started? The Current Growth of Artificial Intelligence, Machine Learning, and Deep Learning | Iraqi Journal For Computer Science and Mathematics	

W https://doi.org/10.52866/ijcsm.2022.01.01.013

12/27	SUBMITTED TEXT	12 WORDS	100% MATCHING TEXT	12 WORDS
	Prediction of Diseases in Smart Health Care System using Machine Learning.		Prediction of Diseases in Smart Health Care System using Machine Learning	

W https://doi.org/10.35940/ijrte.E6482.018520

13/27	SUBMITTED TEXT	22 WORDS	86% MATCHING TEXT	22 WORDS
	S., Sawalha, S., & Abdelnabi, H. (2020). Agile software development: Methodologies and trends. International Journal of Interactive Mobile Technologies, 14(11), 246–270. https://doi.org/10.3991/ijim.v14		S., Sawalha, S., & Abdel-Nabi, H. (2020). Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies (iJIM), 14(11), pp. 246–270. https://doi.org/10.3991/ijim.v14	

W https://doi.org/10.3991/ijim.v14i11.13269

14/27	SUBMITTED TEXT	22 WORDS	100% MATCHING TEXT	22 WORDS
	Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis. International Journal of Applied Business and Management Studies, 6(1), 2021.		Waterfall Process Operations in the Fast-paced World: Project Management Exploratory Analysis. International Journal of Applied Business and Management Studies, 6(1), 91–99.	

SA REPORT-2019294656.docx (D142598150)

15/27	SUBMITTED TEXT 23 WORDS	100% MATCHING TEXT 23 WORDS
	Battineni, G., Sagaro, G. G., Chinatalapudi, N., & Amenta, F. (2020). Applications of machine learning predictive models in the chronic disease diagnosis.	Battineni, G.; Sagaro, G.G.; Chinatalapudi, N.; Amenta, F. Applications of Machine Learning Predictive Models in the Chronic Disease Diagnosis.

W https://doi.org/10.3390/jpm10020021

16/27	SUBMITTED TEXT 13 WORDS	100% MATCHING TEXT 13 WORDS
	th International Conference on Advanced Computing and Communication Systems, ICACCS 2021, 581–585.	th International Conference on Advanced Computing and Communication Systems (ICACCS);

W https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/

17/27	SUBMITTED TEXT 19 WORDS	100% MATCHING TEXT 19 WORDS
	The Role of Machine Learning Algorithms for Diagnosing Diseases. Journal of Applied Science and Technology Trends, 2(01), 10– 19.	The Role of Machine Learning Algorithms for Diagnosing Diseases | Journal of Applied Science and Technology Trends

W https://doi.org/10.38094/jastt20179

18/27	SUBMITTED TEXT 16 WORDS	100% MATCHING TEXT 16 WORDS
	A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images.	A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images,

W https://doi.org/10.38094/jastt20179

19/27	SUBMITTED TEXT 19 WORDS	92% MATCHING TEXT 19 WORDS
	Detection of Hepatitis (A, B, C and E) Viruses Based on Random Forest, K-nearest and Naïve Bayes Classifier.	Detection of Hepatitis (A, B, C and E) Viruses Based on Random Forest, K-nearest and Naive Bayes Classifier", 10

W https://doi.org/10.38094/jastt20179

20/27	SUBMITTED TEXT 29 WORDS	86% MATCHING TEXT 29 WORDS
	Kamble, S. D., Patel, P., Fulzele, P., Bangde, Y., Musale, H., & Gaddamwar, S. (2021). Disease Diagnosis System using Machine Learning. Journal of Pharmaceutical Research International, 185–194. https://doi.org/10.9734/jpri/2021/v33i33b31810	Kamble, S. D., Patel, P., Fulzele, P., Bangde, Y., Musale, H. and Gaddamwar, S. (2021) “Disease Diagnosis System using Machine Learning”, Journal of Pharmaceutical Research International, 33(33B), pp. 185–194. doi: 10.9734/jpri/2021/v33i33B31810.

W https://doi.org/10.9734/jpri/2021/v33i33b31810

21/27	SUBMITTED TEXT	29 WORDS	59% MATCHING TEXT	29 WORDS

https://doi.org/10.1016/j.ihj.2020.04.001 Kumar, N., Narayan Das, N., Gupta, D., Gupta, K., & Bindra, J. (2021). Efficient Automated Disease Diagnosis Using Machine Learning Models. Journal of Healthcare Engineering, 2021. https://doi.org/10.1155/2021/9983652

https://doi.org/10.1155/2021/9983652 Naresh Kumar, Nripendra Narayan Das, Deepali Gupta, Kamali Gupta, Jatin Bindra, " Efficient Automated Disease Diagnosis Using Machine Learning Models", Journal of Healthcare Engineering, vol. 2021, Article ID 9983652, 13 pages, 2021. https://doi.org/10.1155/2021/9983652

W https://www.hindawi.com/journals/jhe/2021/9983652/

22/27	SUBMITTED TEXT	34 WORDS	85% MATCHING TEXT	34 WORDS

https://doi.org/10.35940/ijitee.F1066.0486S419 Magnavita, N., Tripepi, G., & Di Prinzio, R. R. (2020). Symptoms in health care workers during the covid-19 epidemic. A cross-sectional survey. International Journal of Environmental Research and Public Health, 17(14), 1–15.

https://doi.org/10.1006/jcec.2002.1789 Lubis, A. F. (2009). Health economics. USUpress. Magnavita, N., Tripepi, G., & Di Prinzio, R. R. (2020). Symptoms in health care workers during the COVID-19 epidemic. A cross-sectional survey. International Journal of Environmental Research and Public Health, 17(14), 5218.

W https://www.researchgate.net/publication/361308218_An_Analysis_of_the_Impact_of_the_COVID-19_Pand ...

23/27	SUBMITTED TEXT	38 WORDS	91% MATCHING TEXT	38 WORDS

Miller, S., Gilbert, S., Virani, V., & Wicks, P. (2020). Patients⇔ utilization and perception of an artificial intelligence⇓based symptom assessment and advice technology in a British primary care waiting room: Exploratory pilot study. JMIR Human Factors, 7(3). https://doi.

Miller S, Gilbert S, Virani V, Wicks P Patients' Utilization and Perception of an Artificial Intelligence–Based Symptom Assessment and Advice Technology in a British Primary Care Waiting Room: Exploratory Pilot Study JMIR Hum Factors 2020;7(3):e19713 doi: 10.2196/19713

W https://doi.org/10.2196/19713

24/27	SUBMITTED TEXT	22 WORDS	92% MATCHING TEXT	22 WORDS

Teaching Critical Thinking: A Case for Instruction in Cognitive Biases to Reduce Diagnostic Errors and Improve Patient Safety. In Academic Medicine (

Teaching Critical Thinking: A Case for Instruction in Cognitive Biases to Reduce Diagnostic Errors and Improve Patient Safety : Academic Medicine

W https://doi.org/10.1097/ACM.0000000000002518

25/27	SUBMITTED TEXT	11 WORDS	100% MATCHING TEXT	11 WORDS

International Conference on Advances in Computing, Communication Control and Networking,

International Conference on Advances in Computing, Communication Control and Networking (

SA FINAL REPORT_MUHAMMAD HANIF BIN ADNAN_2022912847.pdf (D184517383)

26/27	SUBMITTED TEXT	11 WORDS	100% MATCHING TEXT	11 WORDS
th International Conference on Advanced Computing and Communication Systems (ICACCS).			th International Conference on Advanced Computing and Communication Systems (ICACCS);	

W https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/

27/27	SUBMITTED TEXT	14 WORDS	82% MATCHING TEXT	14 WORDS
Patient-level performance evaluation of a smartphone-based malaria diagnostic application. Malaria Journal, 22(1). https://doi.org/10.1186/s12936-023- 04446-0			Patient-level performance evaluation of a smartphone-based malaria diagnostic application. Malar J 22, 33 (2023). https://doi.org/10.1186/s12936-023-04446-0	

W https://doi.org/10.1186/s12936-023-04446-0