

Article

Not peer-reviewed version

Feature Selection Using Intelligent Agents for Time Improvement in Medical Diagnosis Systems

[Maria Viorela Muntean](#)^{*}, Andreea Florina Hirceagă, [Matei Vasile Căpîlnaş](#)

Posted Date: 10 October 2025

doi: 10.20944/preprints202510.0721.v1

Keywords: feature selection; intelligent agents; classification model; chatbot; medical diagnosis



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a Creative Commons CC BY 4.0 license, which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Feature Selection Using Intelligent Agents for Time Improvement in Medical Diagnosis Systems

Maria Viorela Muntean *, Andreea Florina Hîrceagă and Matei Vasile Căpîlnaş

Department of Informatics, Mathematics and Electronics, 1 Decembrie 1918, University of Alba Iulia, 510009
Alba Iulia, Romania

* Correspondence: mmuntean@uab.ro

Abstract

Feature selection is an important task in medical applications, given that the dimensionality and numerosity of such datasets is very high. In these cases, time parameter becomes also important, along with classification accuracy, in estimating the performance of a learning model. In this work we propose intelligent agent teams that are capable to automatically discover the best time to build models while keeping the general accuracy at highest levels. For computing attributes' relevance for the classification process, several techniques were used: Information Gain, Gain Ratio, Correlation and Relief Attribute Evaluator. The two proposed agent teams discovered that an optimum subset composed by 20 attributes (out of 133 attributes of the initial dataset) leads to accuracy rates equally to the ones registered on the entire dataset, meaning 98%, using Naive Bayes learning model, while improving the time taken to build model from 0.1 seconds to 0.03 seconds. For the proposed dataset, Naive Bayes outperformed other classification techniques, such as: J48, Random Forest, and D14MlpClassifier. The proposed agents also integrated the best discovered model into a chatbot that performs medical diagnosis based on the symptoms collected from users.

Keywords: feature selection; intelligent agents; classification model; chatbot; medical diagnosis

1. Introduction

It is well known that medical datasets have many features, and learning such data is often time consuming. To deal with this issue, different approaches based feature selection are proposed.

The authors [1] propose the development of an efficient feature selection methodology tailored for stress detection using chest electrodermal activity (EDA) signals. By employing this feature selection technique, the study aims to enhance the performance of the stacking ensemble model, ensuring that the most relevant features are utilized for accurate stress classification. This approach not only improves the model's accuracy but also reduces computational complexity, making it more suitable for real-time applications in wearable devices.

The study [2] introduces a hybrid feature selection approach that combines both filter and wrapper methods to identify the most relevant clinical features for diabetes prediction. Initially, statistical measures such as correlation coefficients are employed to eliminate irrelevant and redundant features. Subsequently, a wrapper-based method using a Genetic Algorithm (GA) is applied to select an optimal subset of features that enhance the performance of the machine learning classifiers. This two-tiered feature selection strategy effectively reduces the dimensionality of the dataset, mitigates the risk of overfitting, and improves the predictive accuracy of the models.

In the paper [3], the authors propose the integration of explainable AI methods to discern and validate the most relevant features (biomarkers) for NAFLD diagnosis. By employing ensemble machine learning models alongside XAI tools, the system not only achieves high diagnostic accuracy (over 95%) but also provides transparency in its decision-making process. This approach facilitates the identification of critical clinical attributes, enhancing the interpretability and trustworthiness of the AI system in a medical context.

The authors [4] propose the development of an embedded feature selection method that synergistically combines filter and wrapper techniques. This hybrid approach aims to balance computational efficiency with predictive accuracy, addressing the challenges posed by high-dimensional clinical datasets.

The paper [5] introduces a novel metaheuristic algorithm, the Memory-Based Sand Cat Swarm Optimizer (MSCSO), specifically designed to enhance feature selection processes in medical diagnostic applications. By employing a wrapper methodology, the algorithm evaluates subsets of features based on the performance of a specific classifier, ensuring that the selected features contribute significantly to the model's predictive accuracy.

The SOFIA method proposed in [6] is applied to various medical datasets, demonstrating its effectiveness in reducing dimensionality while maintaining or enhancing classification accuracy. This is particularly beneficial in medical diagnostics, where datasets often contain numerous features, some of which may be irrelevant or redundant.

The article [7] investigates the significance of selecting relevant features from high-dimensional medical datasets to enhance the performance of machine learning models in diagnosing heart disease. The findings indicate that even with a reduced number of features, the performance of classification models improved notably, including enhanced accuracy and reduced training time, compared to models trained on the full feature set.

Other medical applications [8] focus on developing a non-contact, rapid, and reliable method for detecting elevated body temperature in elderly individuals—a critical indicator of potential viral infections. The study utilizes thermal imaging and deep learning techniques to enhance early detection capabilities in healthcare settings.

Generative AI (GAI) chatbots [9] are also used in various domains, including medical domain, where some studies introduce novel decision-making framework designed to assist organizations.

In previous studies [10,11], we proposed multi-agent systems manages various machine learning tasks—from data preprocessing and labeling to model selection and evaluation—by delegating them to specialized agents.

Other studies use multi-agent systems for evaluation of a portable, intelligent device designed to assess and treat Cervical Motor Control (CMC) impairments [12], or to integrates physical activity with interactive learning to enhance motor and cognitive development among school and high school students. [13]

To automatically find the optimal subset for learning medical data, we propose a multi-agent system that integrates feature selection and classification tasks. In the preprocessing stage, a Feature Selection Agent communicates with three subagents, namely Attribute Evaluator Subagent, Search Method Subagent and Threshold Subagent to find the optimum feature subset, for keeping the accuracy at high values rates while automatically finding the best time for building models. The optimum learning model, returned by a Classification Agent, with two subagents (Model Selection Agent and Model Performance Subagent), is integrated into a chatbot system and proved its confidence for medical diagnosis. The proposed model is trained on binary data (indicating the presence or absence of each known symptom in the user's list), and the next step will be testing models on text data, as performed in the research described in [14].

2. Materials and methods

Intelligent Agents Based on Feature Selection for Time Improvement

The proposed system architecture is suitable for time improvement in learning medical data. The system is composed of two main agents and five subagents, with specific tasks (Figure 1). The proposed agents and subagents were implemented using Weka data mining tool [15,16], and Jade platform [19].

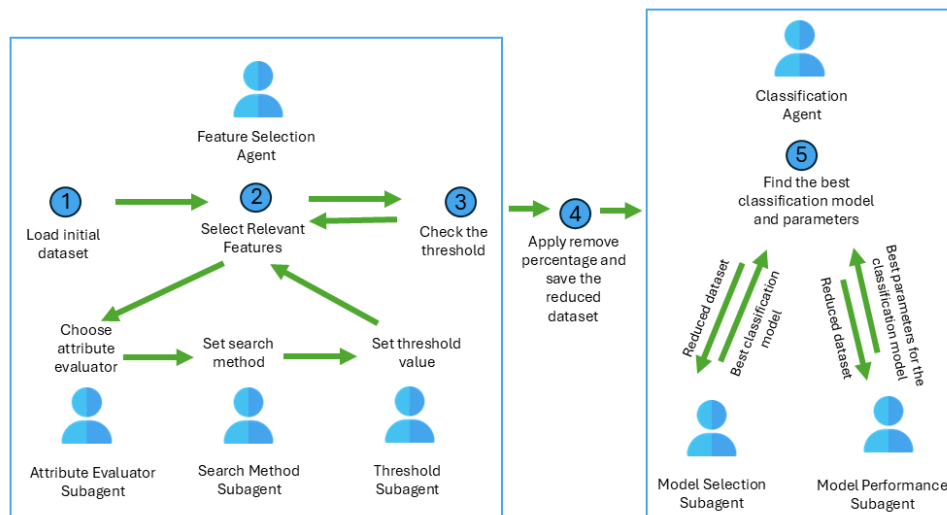


Figure 1. Intelligent agents' architecture.

The proposed agents collaborate to automate the process of finding the best subset and the best model for medical applications. Agent behaviors are presented below.

Feature Selection Agent:

- loads the initial medical dataset;
- finds the optimum subset in terms of accuracy and time taken to build models;
- if the threshold is reached, then sends the reduced dataset to Classification Agent.

Classification Agent:

- receives the reduced dataset from Feature Selection Agent;
- finds the best model and the best model configuration;
- sends the final model to the chatbot to be used for medical diagnosis.

The proposed system contains five subagents for time and model performance improvement. Their specific behaviors are described below.

Attribute Evaluator Subagent:

- applies different evaluation methods to compute the relevance of an attribute (InfoGain, GainRatio, Correlation, ReliefF);

Search Method Subagent:

- uses search method (Ranker) for finding the relevant features that will be included in a specific subset of data;

Threshold Subagent:

- applies different threshold values to decide if the attribute is relevant and will be kept in the subset.

Model Selection Subagent:

- learns the medical data with different models (decision trees, naïve bayes, deep learning neural networks);
- chooses the best model considering the model performance in terms of accuracy and time.

Model Performance Subagent:

- finds the best model configuration;
- send the optimum model to the system's chatbot to be used in medical diagnosis.

For selecting the best model, the autonomous subagents collaborate to find an optimum subset of attributes with the best classification time values, while the general accuracy is kept on its initial rate, or it is even improved. The first agent's task (Attribute Evaluator Subagent) is to evaluate the dataset with different attribute evaluators (InfoGainAttributeEval, GainRatioAttributeEval, CorrelationAttributeEval, ReliefFAttributeEval) to determine the importance of each attribute for the classification process. The second agent (Search Method Subagent) uses different search methods

(BestFirst, GreedyStepwise, Ranker) to decide which of the discovered subsets of attributes best classifies (in terms of time and accuracy) the dataset. The third agent (Threshold Subagent) performs a final refinement of the optimal subset as proposed in [20], by eliminating the attributes that have the importance score below a threshold. One of this paper contributions' is that the proposed Threshold Subagent automatically finds the best threshold for attributes' relevance by evaluating the dataset with different thresholds values. The preprocessing stage is monitored by the Feature Selection Agent, that collects messages from its subagent team and sends the optimum discovered feature subset to the classification module.

The classification module uses different classification models to learn the data (Model Selection Subagent) to decide which is the most suitable one for the considered dataset. For the previously chosen model, the Model Performance Subagent performs the classification task using different parameter values, to find the optimum model configuration. The final model (trained and tested on the entire dataset) is sent to Classification Agent and used for predicting the disease for new registered patients.

As a remark, the proposed system optimizes the entire process (from preprocessing to classification stages) without any human support, being suitable for real-time diagnosis of different diseases. The preprocessing and classification results, together with the dataset description for the medical domain are presented in the fourth section.

Medical Diagnosis System Using the Best Discovered Learning Model

The proposed system contains a chatbot that was built using the Rasa framework [21] and it was designed to interact with users in a medical context. This framework recognizes the users' intents extracts the entities from their messages. For each intent, several examples of possible expressions are defined, which are used to train the Natural Language Processing model to recognize those intentions in real conversations. Also, symptom entities are defined, which allow the extraction of symptoms from the users' messages, along with synonyms for normalizing terms.

Sending symptoms (Figure 2):

- The user sends the symptoms via POST/webhook;
- The Flask Server forwards the POST/webhook/rest to the Rasa Server;
- Race Server runs action_provide_diagnosis_and_suggest:
 - Checks if the symptom is valid.
 - ✓ If it's valid, add it to the list and suggest other symptoms.
 - ✓ Otherwise, it returns a message that the symptom has not been recognized.
- A JSON response is sent to the Web Site with the symptoms and symptom suggestions.

The user says they have no symptoms (Figure 2):

- The user sends the message "No, I don't have any symptoms" via POST/webhook.
- A new POST/webhook/rest call is made to the Race Server.
- action_provide_diagnosis shall be executed, which:
 - Sends the data to the model for prediction.
 - Gets a diagnosis.
 - Replies with a JSON message containing the diagnostic results.

Conversational rules were defined, meaning the steps that the bot follows when it identifies a specific intent. For example, if the user greets, the bot will respond with a welcome message. If the user declares that they are sick, the bot will ask for details about the symptoms. If symptoms are mentioned, a personalized action is triggered, which has the role of analyzing the symptoms and providing a relevant suggestion or a possible diagnosis.

The classification model obtained in the previous stage is trained allows that chatbot to predict a diagnosis based on user-reported symptoms. The main goal is to interpret the symptoms expressed in natural language, to correlate them with a database of known symptoms stored in MySQL database management system [22], and to predict a medical diagnosis using the previously trained model (Figure 2).

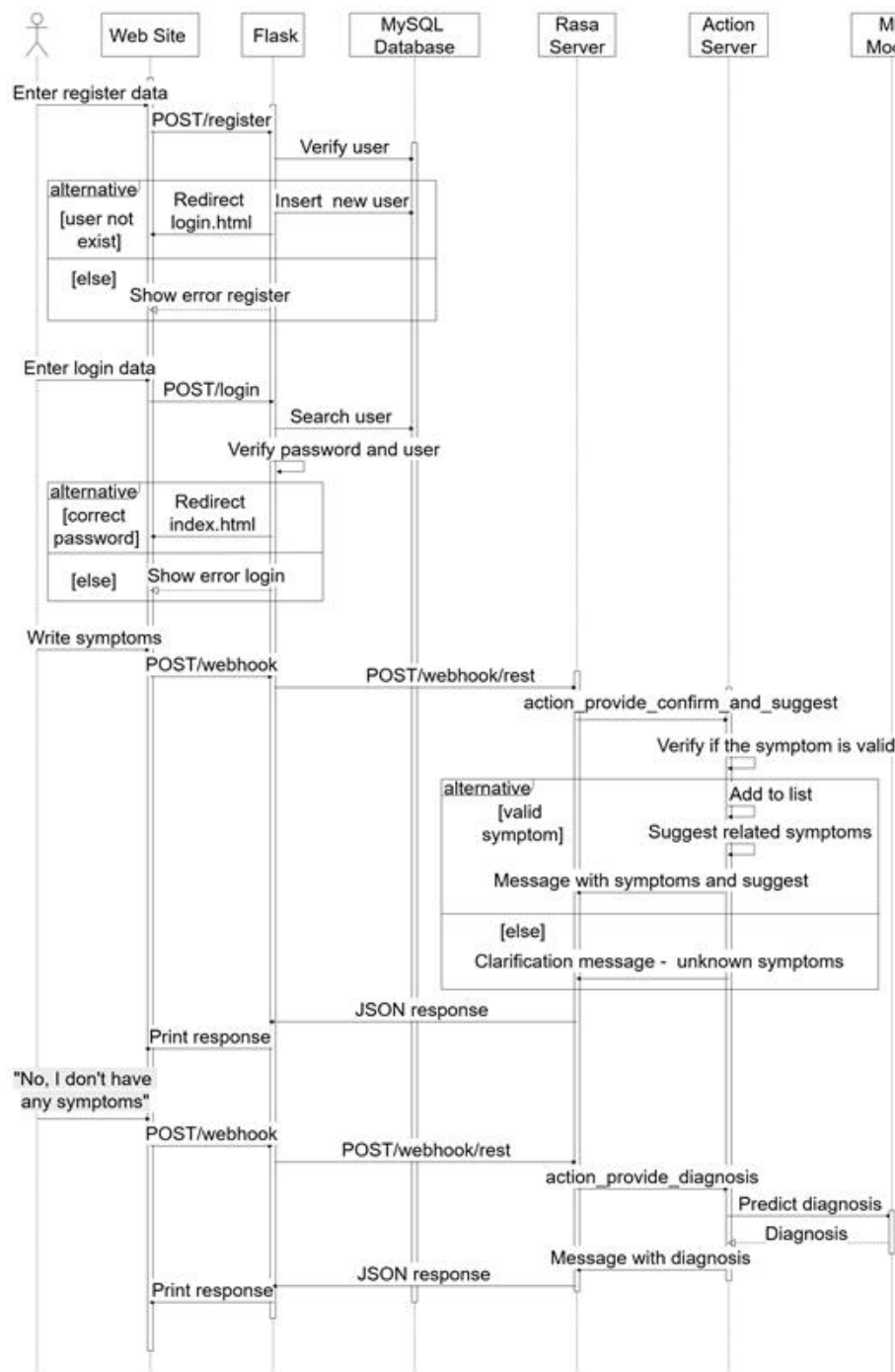


Figure 2. Medical diagnosis system architecture.

In this project, the key components of the conversational assistant developed with Rasa were extended and configured, with a focus on detecting symptoms and adapting the conversation according to the users' state (Figure 3).

If the user enters symptoms exactly as they appear in the database (for example: headache, vomiting, abdominal pain) and synonyms written as in the system file, the system recognizes them immediately and with high confidence. In this case: the text is converted to a standard format (lowercase letters, without diacritics); check if the symptom name (e.g., abdominal pain) exists as a column name in the dataset; if it exists, the symptom is added directly to the list of recognized symptoms; no fuzzy matching required.

If the user enters symptoms and synonyms in another form (typographical errors, e.g., *skin ras* instead of *skin rash*, *no apetite* instead of *no appetite*), the chatbot uses a fuzzy matching mechanism to interpret them correctly.

If a symptom is entered by the user in a slightly wrong form but similar to the name in the database, it is directly compared to the column names from the database. If the fuzzy matched score exceeds the threshold of 80, the symptom is considered valid and is added to the list of recognized symptoms.

Conversely, if the user enters a misspelled synonym compared to the thesaurus format, it is compared to the list of synonyms defined in the synonym mapping structure. If the fuzzy matched score between the entered synonym and a known synonym is greater than 85, then the system maps it to the primary symptom associated with that synonym. This symptom is then added to the final list of symptoms that will be used for suggestions or diagnosis.

When collecting data from the user, the system constructs a binary vector (with 0 and 1) indicating the presence or absence of each known symptom in the user's list. This vector is transmitted to the Naïve Bayes model to predict the diagnosis. The model returns both the prediction and its probability, which allows to include a confidence value in the final answer. The message sent by the bot contains the predicted diagnosis, the list of symptoms on which the prediction was based, and the confidence level of the model.

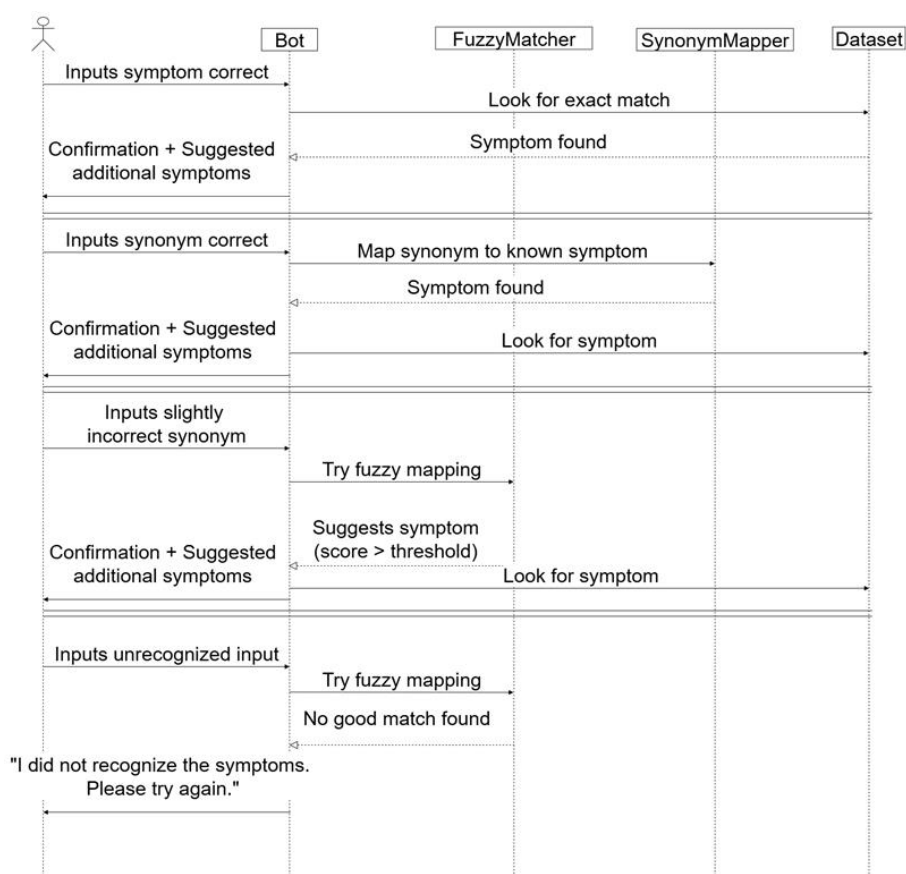


Figure 3. The personalized chatbot architecture.

In the Natural Language Understanding Model, the intentions of user sick, user alright and symptoms mention have been defined and completed, aimed at capturing the state of health expressed by the user. The symptoms mention intent has been significantly expanded, including many of examples covering a wide range of symptoms, each appropriately labeled with the symptom entity. Also, several synonyms for symptoms have been implemented, making it easier to recognize them.

we Regarding the conversational rules, these have been defined for automatic replies to frequent intents. Specific rules have been introduced for managing symptoms and user status, as well as for triggering custom actions (e.g., action provide diagnosis, action provide diagnosis and suggest).

The conversational domain was modelled, so the symptom entity has been defined and associated with two slots (symptoms and suggested symptoms) to allow the collection and processing of received symptoms. In the proposed system, several relevant responses were added to basic medical and interaction scenarios and the personalized actions needed to process symptoms and provide an appropriate response have been integrated into the conversational logic.

3. Experimental Results

The Symptom-Disease Prediction Dataset used for validating the proposed systems was published in 2024 and was obtained from [23,24] and stores a collection of symptoms for various diseases. The data was collected from reputable medical literature, clinical observations, and expert consensus, and many attributes can be also collected from sensors in real-time applications.

3.1. Symptom-Disease Prediction Dataset Description

A sample of initial collected data is presented in Figure 4. The dataset contains 133 attributes and 4961 instances. The attributes' values can be 1 or 0 meaning the symptom is present or absent.

fluid_over_blood_in_prominen	palpitatio	painful_w_pus_filled	blackhear	scurring	skin_peel	silver_like	small_der	inflamma	blister	red_sore_yellow_cri	prognosis
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Fungal Infection
0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0 Allergy
0	0	0	0	0	0	0	0	0	0	0	0 Allergy

Figure 4. Sample of initial data.

The class attribute can have one of the following 41 values: Acne, Urinary Tract Infection, Psoriasis, Impetigo, Fungal Infection, Allergy, GERD, Chronic Cholestasis, Drug Reaction, Peptic Ulcer Disease, AIDS, Diabetes, Gastroenteritis, Bronchial Asthma, Hypertension, Migraine, Cervical Spondylosis, Paralysis (brain hemorrhage), Jaundice, Malaria, Chickenpox, Dengue, Typhoid, Hepatitis A, Hepatitis B, Hepatitis C, Hepatitis D, Hepatitis E, Alcoholic Hepatitis, Tuberculosis, Common Cold, Pneumonia, Dimorphic Hemorrhoids (piles), Heart Attack, Varicose Veins, Hypothyroidism, Hyperthyroidism, Hypoglycemia, Osteoarthritis, Arthritis, Vertigo.

3.2. Feature Selection and Classification

For feature selection and classification, the Weka Machine Learning Tool was used [15–18].

In the dataset preprocessing stage, we used feature selection methods to reduce the size of the dataset and keep only the attributes relevant to classification process. We have applied several Attribute Evaluators: InfoGainAttributeEval (that computes the information gain related to the class attribute for a certain attribute) [16], GainRatioAttributeEval (that evaluates the importance of an attribute by computing its gain ratio related to the class attribute) [16], CorrelationAttributeEval (that measures the Pearson's correlation between an attribute and the class attribute to establish the attribute's importance) [16], ReliefFAttributeEval (that measures the importance of an attribute by repeatedly selecting instances and examining the attribute's value in the nearest neighbors belonging to the same and different classes) [17,18]. The attributes are selecting according to their importance (ranging from 0 to 1) and the attributes that are above a specified threshold are selected for the optimal subset.

Figure 6. Sample of preprocessed data.

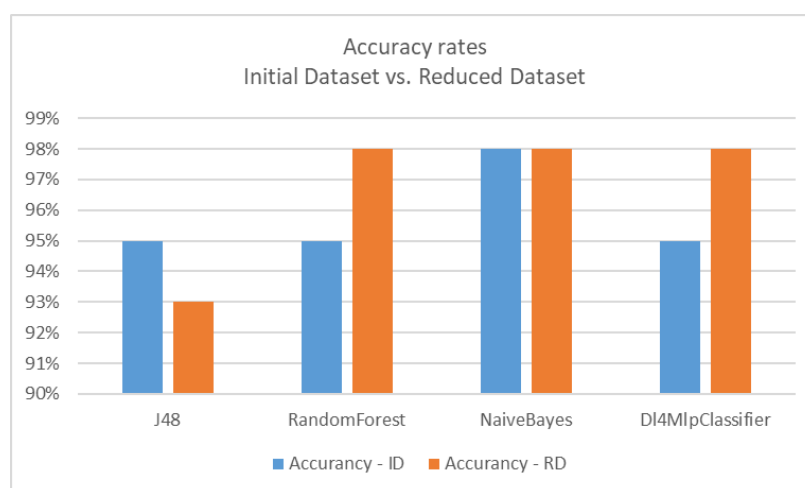
In the classification stage, the initial and reduced dataset was learned by different classifiers (J48, RandomForest, NaiveBayes, D14MlpClassifier) to decide which is the best model for the considered data and to highlight the time improvement in the reduced data.

In the case of the Random Forest and Deep learning neural network (D14MlpClassifier) classifiers, the preprocessed set significantly improved the testing accuracy, increasing from 95% to 98%. Naive Bayes maintained a high and stable accuracy of 98%. Only in the case of Decision Tree (J48) was registered a slight decrease in performance on the testing dataset (from 95% to 93%), but it remains an acceptable value (Table 1, Figure 7). For testing the models, 20% split percentage method was used, meaning that 80% of instances (3969 patients) were used to train the models, and the remaining 20% of instances (992 patients) were used to test the discovered models.

Preprocessing the dataset by feature selection (Information Gain with threshold equal to 0.02) did not negatively affect the performance of the models. On the contrary, in some cases it increased the accuracy rates in the testing phase, which suggests better generalization and reduced overfitting. The preprocessed dataset is therefore more efficient than the initial dataset. Naïve Bayes classifier was chosen to be integrated into the proposed application, because the model achieved an accuracy of 98% on the test set, demonstrating a better generalization on the new data. Also, the model based on Naive Bayes was built in 0.03 seconds, being the best performing model (Table 2, Figure 8).

Table 1. Classification accuracy for different learning models.

Classifier	Accuracy on Initial Dataset (ID)	Accuracy on Reduced Dataset (RD)
J48 (Decision tree)	95%	93%
RandomForest	95%	98%
NaïveBayes	98%	98%
D14MlpClassifier (Deep learning)	95%	98%

**Figure 7.** Classification accuracy for different models.**Table 2.** Time taken to build models.

Classifier	Time for Initial Dataset (ID)-seconds	Time for Reduced Dataset (RD)-seconds
J48 (Decision tree)	0.71	0.08
RandomForest	1.13	0.33
NaïveBayes	0.1	0.03

Dl4MlpClassifier (Deep learning)	28.26	0.24
----------------------------------	-------	------

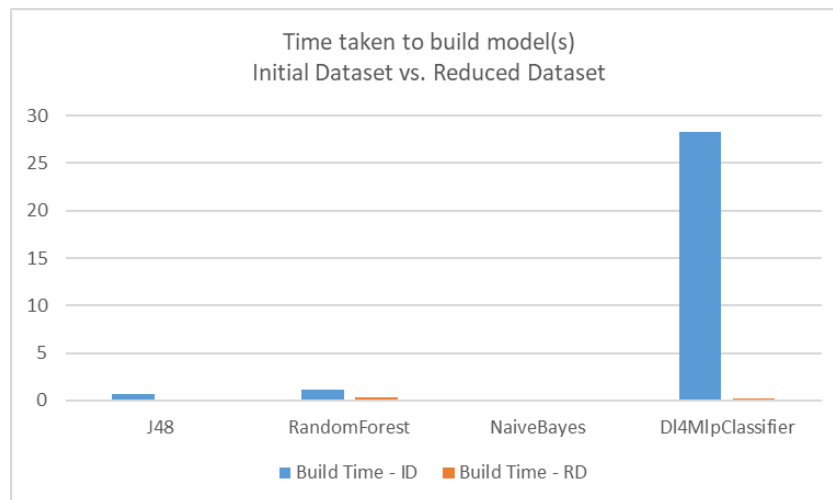


Figure 8. Time taken to build models.

3.3. Testing the Chatbot Based Naïve Bayes Classifier

For implementing the proposed system, the Rasa framework was used [21].

Communication between the chat interface and the Rasa model is done through a webhook defined in Flask [25], which sends the user's messages to Rasa and displays their responses in the chat (Figure 9 and Figure 10). Thus, every interaction with the chatbot is associated with the authenticated account.

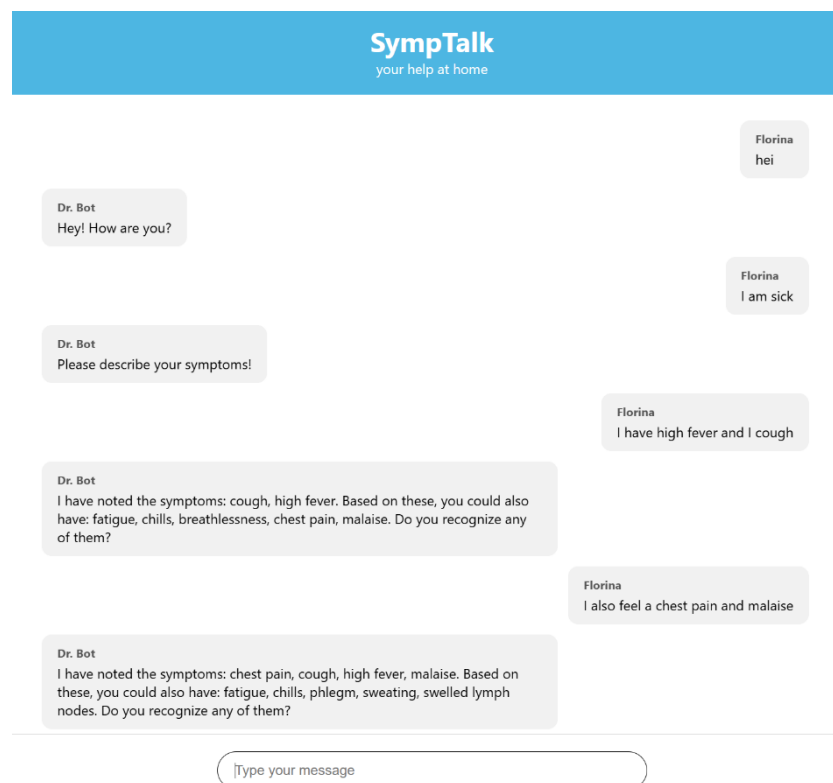


Figure 9. Chatbot for medical diagnosis based on Naïve Bayes model.

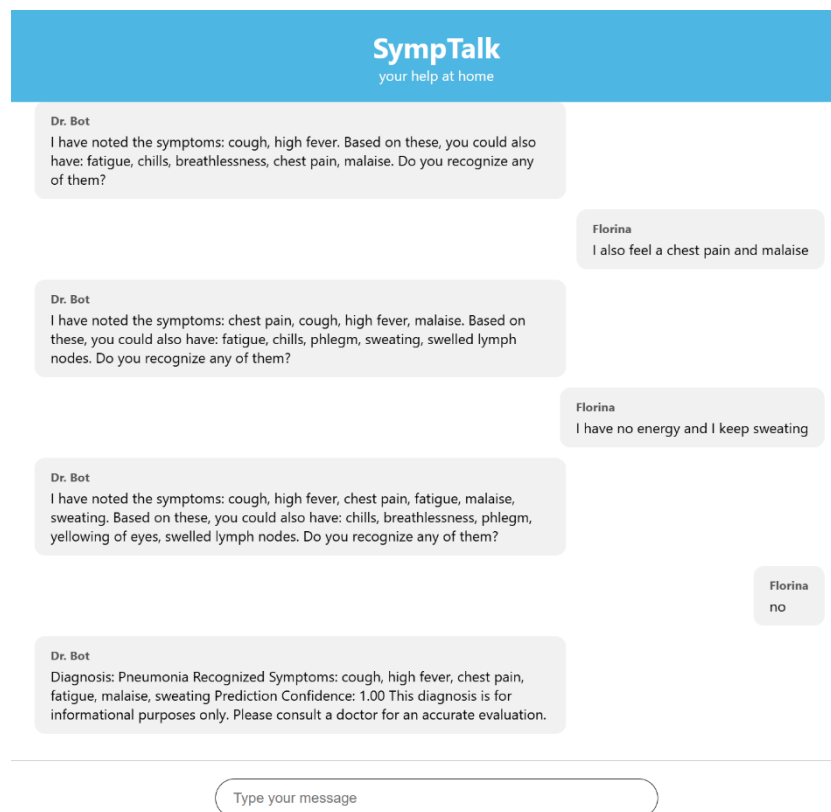


Figure 10. Chatbot for medical diagnosis based on Naïve Bayes model.

Webhook flow (Figure 9 and Figure 10):

1. The user sends a message in the interface
2. The JavaScript in the index.html makes a POST to /webhook with the message
3. Flask takes the message and sends it on to Rasa
4. The race analyzes the message, executes actions
5. The race sends a response (JSON) to the Flask
6. Flask returns the response to the frontend
7. The frontend displays it in chat as a bot message

The main goal is to interpret symptoms expressed in natural language, correlate them with a database of known symptoms, and predict a medical diagnosis using the previously trained Naive Bayes model.

4. Discussion

In this work, we have shown that intelligent agents can perform the automation task, being able to discover time taken to build model equal to 0.03 seconds in the case of Naïve Bayes classifier, while keeping the general accuracy at 98% (as discovered also in other studies on the same data [26,27]). In our study, the optimum feature set was discovered using Information Gain evaluation method, and Ranker search method, with a threshold of selecting attributes equal to 0.02. The reduced dataset contained 20 relevant attributes out of 133 attributes of the initial dataset. The proposed system automates the process of finding the optimum learning time, while keeping the accuracy at high levels.

5. Conclusions and Future Developments

The optimum discovered model was automatically integrated into a medical diagnosis application and used by a chatbot in establishing the diagnosis for different diseases. We mention

that the proposed system can help the patients with information, but the final decision belongs to the doctors.

As further development, we plan to test the Naïve Bayes model also on text data, meaning considering not just the presence or the absence of a symptom, but considering also the description of the symptoms in the training process.

Funding: This research received financial support from the funds for scientific research of 1 Decembrie 1918 University of Alba Iulia, Romania.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in <https://data.mendeley.com/datasets/dv5z3v2xyd/1> (accessed on 19 May 2025).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Almadhor, A.; Sampedro, G.A.; Abisado, M.; Abbas, S. Efficient Feature-Selection-Based Stacking Model for Stress Detection Based on Chest Electrodermal Activity. *Sensors* 2023, 23, 6664. <https://doi.org/10.3390/s23156664>
2. Haq, A.U.; Li, J.P.; Khan, J.; Memon, M.H.; Nazir, S.; Ahmad, S.; Khan, G.A.; Ali, A. Intelligent Machine Learning Approach for Effective Recognition of Diabetes in E-Healthcare Using Clinical Data. *Sensors* 2020, 20, 2649. <https://doi.org/10.3390/s20092649>
3. Pludowski, J.; Mulawka, J. Application of Artificial Intelligence in Support of NAFLD Diagnosis. *Appl. Sci.* 2024, 14, 10237. <https://doi.org/10.3390/app142210237>
4. Alyasin, E.I.; Ata, O.; Mohammedqasim, H.; Mohammedqasem, R. Enhancing Self-Care Prediction in Children with Impairments: A Novel Framework for Addressing Imbalance and High Dimensionality. *Appl. Sci.* 2024, 14, 356. <https://doi.org/10.3390/app14010356>
5. Qtaish, A.; Albashish, D.; Braik, M.; Alshammari, M.T.; Alreshidi, A.; Alreshidi, E.J. Memory-Based Sand Cat Swarm Optimization for Feature Selection in Medical Diagnosis. *Electronics* 2023, 12, 2042. <https://doi.org/10.3390/electronics12092042>
6. Bravo, F.P.; García, A.A.D.B.; Russo, L.M.S.; Ayala, J.L. SOFIA: Selection of Medical Features by Induced Alterations in Numeric Labels. *Electronics* 2020, 9, 1492. <https://doi.org/10.3390/electronics9091492>
7. Pathan, M.S.; Nag, A.; Pathan, M.M.; Dev, S. Analyzing the Impact of Feature Selection on the Accuracy of Heart Disease Prediction. *Healthcare Analytics* 2022, 2, 100060. <https://doi.org/10.1016/j.health.2022.100060>
8. Ghourabi, M.; Mourad-Chehade, F.; Chkeir, A. Eye Recognition by YOLO for Inner Canthus Temperature Detection in the Elderly Using a Transfer Learning Approach. *Sensors* 2023, 23, 1851. <https://doi.org/10.3390/s23041851>
9. Ilieva, G. Extension of Interval-Valued Hesitant Fermatean Fuzzy TOPSIS for Evaluating and Benchmarking of Generative AI Chatbots. *Electronics* 2025, 14, 555. <https://doi.org/10.3390/electronics14030555>
10. Muntean, M.V. Real-Time Detection of IoT Anomalies and Intrusion Data in Smart Cities Using Multi-Agent System. *Sensors* 2024, 24, 7886. <https://doi.org/10.3390/s24247886>
11. Muntean, M.V. Multi-Agent System for Intelligent Urban Traffic Management Using Wireless Sensor Networks Data. *Sensors* 2022, 22, 208. <https://doi.org/10.3390/s22010208>
12. Sales Mendes, A.F.; Sánchez San Blas, H.; Pérez Robledo, F.; De Paz Santana, J.F.; Villarrubia González, G. A Novel Multiagent System for Cervical Motor Control Evaluation and Individualized Therapy: Integrating Gamification and Portable Solutions. *Multimedia Systems* 2024, 30, 131. <https://doi.org/10.1007/s00530-024-01328-6>
13. Sánchez San Blas, H.; Sales Mendes, A.; de la Iglesia, D.H.; Silva, L.A.; Villarrubia González, G. A Multiagent Platform for Promoting Physical Activity and Learning through Interactive Educational Games

- Using the Depth Camera Recognition System. *Entertainment Computing* 2024, 49, 100629. <https://doi.org/10.1016/j.entcom.2023.100629>
14. Chen, X., Yi, H., You, M. et al. Enhancing diagnostic capability with multi-agents conversational large language models. *npj Digit. Med.* 8, 159 (2025). <https://doi.org/10.1038/s41746-025-01550-0>
 15. <https://ml.cms.waikato.ac.nz/weka> (accessed on 19 May 2025)
 16. Eibe Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for “Data Mining: Practical Machine Learning Tools and Techniques”, Morgan Kaufmann, Fourth Edition, 2016.
 17. Kenji Kira, Larry A. Rendell: A Practical Approach to Feature Selection. In: Ninth International Workshop on Machine Learning, 249-256, 1992.
 18. Marko Robnik-Sikonja, Igor Kononenko: An adaptation of Relief for attribute estimation in regression. In: Fourteenth International Conference on Machine Learning, 296-304, 1997.
 19. <https://jade.tilab.com/> (accessed on 19 May 2025)
 20. Maria Muntean, Honoriu Vălean, Remus Joldeș, Emilian Ceuca, Feature selection for classifier accuracy improvement, *Acta Universitatis Apulensis*, No. 26/2011 pp. 203-216.
 21. <https://legacy-docs-oss.rasa.com/docs/rasa/> (accessed on 19 May 2025)
 22. <https://www.mysql.com/> (accessed on 19 May 2025)
 23. Tucker, Jay (2024), “SymbiPredict”, Mendeley Data, V1, doi: 10.17632/dv5z3v2xyd.1
 24. <https://data.mendeley.com/datasets/dv5z3v2xyd/1> (accessed on 19 May 2025)
 25. <https://flask.palletsprojects.com/en/stable/> (accessed on 19 May 2025)
 26. Mama, Eden & Sheri, Liel & Aperstein, Yehudit & Apartsin, Alexander. (2025). From Fuzzy Speech to Medical Insight: Benchmarking LLMs on Noisy Patient Narratives. 10.48550/arXiv.2509.11803.
 27. Leila Aissaoui Ferhi & Manel Ben Amar & Atef Masmoudi & Fethi Choubani & Ridha Bouallegue, 2025. “Improving Symptom-Based Medical Diagnosis Using Ensemble Learning Approaches,” *Systems Research and Behavioral Science*, Wiley Blackwell, vol. 42(4), pages 1294-1321, July.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.