Healthcare Data Mining for Disease Prediction

Laman Panakhova, Shahla Azizova, Khadija Ahmadova, Mahbuba Jafarzada, Emil Hajiyev
Intro to Big Data Analytics (INFT-4836)
School of Information Technologies and Engineering, ADA University, Baku, Azerbaijan
Emails: {lpanakhova16882, sazizova11542, kahmadova12720, mjafarzada14099, ehajiyev16259}@ada.edu.az

***Abstract*—This project is developed as part of the requirements for the course *Intro to Big Data Analytics (INFT-4836)*. The primary objective is to apply big data analytics techniques to the healthcare domain, focusing on disease prediction through data mining methods. The project will include data acquisition, preprocessing, exploratory data analysis, and the use of scalable big-data technologies such as PySpark and distributed file systems. In addition, machine learning algorithms will be applied to uncover meaningful patterns and generate predictive insights from large healthcare datasets, ultimately supporting better decision-making and improved healthcare outcomes.**

***Index Terms*—Big Data Analytics, Healthcare, Disease Prediction, Data Mining, PySpark, Machine Learning, Distributed Systems**

I. Introduction

Big data analytics has become really important in today's world because it helps people make sense of huge amounts of information. In this project, we wanted to use what we learned about big data in class and apply it to something real, specifically, healthcare. Hospitals, wearable devices, and medical tools are creating tons of data every day, but having data alone isn't enough. We need ways to analyze it and find patterns that can actually help doctors make decisions, improve treatments, and help patients get better care.

Research shows that data mining can be really useful for spotting trends in diseases and predicting health risks before they get serious [1]. By using these techniques, healthcare workers can make smarter decisions because they have useful information instead of just raw numbers. For example, Palaniappan and Awang [2] showed that by using patient info like age, blood pressure, and cholesterol, predictive models could successfully identify who might have heart problems. This shows how computers and algorithms can help doctors spot diseases early.

Other studies also support this. Soni et al. [3] found that machine learning can figure out which factors affect diseases the most and sometimes work better than traditional methods. Vijiyarani and Sudha [4] looked at lots of techniques, like classification, clustering, and association rules, and found that these can help predict diseases and also understand how diseases are connected.

With new big data tools, it's now possible to handle even larger datasets. Kanakaraddi et al. [5] suggested combining data mining with machine learning and distributed computing to make disease prediction systems faster and more accurate. These systems can analyze data in real-time, which is really useful in healthcare.

In this project, we try to do something similar: use big data tools like MapReduce and distributed systems to analyze healthcare data and predict diseases. By combining machine learning with scalable data processing, we hope to find patterns that could help doctors make better decisions and improve patient care. The main goal is to show that big data analytics can turn raw medical data into real, useful information for healthcare.

II. Project Scope and Objectives (15%)

A. Scope

This project focuses on the healthcare field, specifically on predicting diseases using a large collection of patient symptom data. For this work, we use the publicly available *Disease Prediction Using Machine Learning* dataset from Kaggle.[1] The dataset contains 4,920 patient entries with 41 symptom-based features, along with one final column that indicates the diagnosed disease.

Each symptom (such as *itching, skin rash, vomiting, fatigue, headache, chest pain*) is stored as a binary value, where 1 means the symptom is present and 0 means it is not. The target column, *prognosis*, includes diseases like diabetes, migraine, jaundice, arthritis, hypertension, and several others. Because the dataset is fairly large and high dimensional, it is suitable for applying big data tools such as PySpark for distributed analysis, machine learning, and data exploration.

B. Objectives

The main goal of this project is to explore how big data analytics, specifically PySpark, can be used in healthcare to build a scalable and efficient disease prediction model. The project involves distributed preprocessing, exploratory data analysis, and training a Random Forest classifier using Spark MLlib.

The specific objectives are:

- To preprocess, clean, and analyze a high-dimensional healthcare dataset using PySpark to show how large datasets can be handled efficiently.
- To perform exploratory data mining, such as examining symptom frequencies, disease counts, and relationships between symptoms and diseases.
- To build and train a Random Forest classifier using Spark MLlib, including cross-validation and hyperparameter tuning.

[1] Dataset Link

- To evaluate the model using accuracy, precision, recall, F1-score, and confusion matrices.
- To study feature importance scores to understand which symptoms have the most influence on the final prediction.

1) Justification of Application Area: Healthcare generates massive amounts of data from electronic health records, lab results, medical devices, and more. Traditional tools often struggle to process such large and diverse data efficiently. Big data frameworks like PySpark allow distributed processing, faster computation, fault tolerance, and scalable machine learning workflows.

Disease prediction is an important application in healthcare because early identification can help reduce treatment delays and improve patient outcomes. Using big data tools for disease prediction allows large-scale symptom analysis, faster pattern detection, and supports better decision-making for healthcare systems.

2) Research Questions: This project aims to answer the following research questions:

1) How effectively can PySpark improve the scalability and speed of preprocessing and analyzing healthcare data?
2) How well does a Random Forest classifier perform on high-dimensional, symptom-based disease prediction when trained using Spark MLlib?
3) Which symptoms appear most often across the dataset, and what patterns can be discovered through distributed exploratory data mining?
4) Based on feature-importance results, which symptoms have the strongest influence on predicting diseases?
5) What limitations arise from using only binary symptom features for multi-class disease prediction?

C. Dataset Column Description

The dataset is made up of 41 binary symptom columns and one target column, *prognosis*. Each symptom is stored as either 1 (symptom present) or 0 (symptom absent). These columns are used during both training and testing of the PySpark models.

Some examples of important symptom features include:

- itching: indicates skin irritation.
- skin_rash: visible rash or discoloration.
- continuous_sneezing: repetitive sneezing.
- fatigue: general tiredness.
- joint_pain: stiffness or pain in joints.
- vomiting: occurrence of nausea and vomiting.
- high_fever: elevated temperature.
- headache: persistent or severe head pain.
- breathlessness: difficulty breathing.
- chest_pain: discomfort or tightness in the chest.

The target column, prognosis, contains the disease labels. Before training the model, we convert this column into numeric form using Spark's StringIndexer. All steps, from preprocessing to prediction, are carried out in PySpark to ensure that the workflow can be easily scaled to much larger datasets. A summary of the dataset, including the number of samples and features, is shown in Table I.

TABLE I
DATASET OVERVIEW

Property	**Training Set**	**Testing Set**
Number of Samples	4920	1230
Number of Symptom Features	132	132
Label Column	prognosis	prognosis
Feature Type	Binary (0/1)	Binary (0/1)
Missing Values	None	None
Data Format	CSV	CSV

III. RELATED WORK

Many researchers have studied how data mining and machine learning can help predict diseases. For example, Bhatla and Jyoti [1] found that neural networks gave the best accuracy for predicting heart disease compared to other traditional algorithms. Their work also showed that picking the right features and designing the model carefully is really important for good predictions.

Bhatnagar et al. [2] showed that proper preprocessing of data along with supervised learning methods can greatly improve diagnostic results. They also suggested that combining techniques like clustering, feature selection, and classification makes medical prediction systems more reliable.

Srinivas et al. [3] focused on clinical features such as chest pain, cholesterol, and blood pressure and found these to be strong indicators of heart disease risk. This supports the idea that symptom-based datasets can be useful for early detection.

Some studies have looked at predicting multiple diseases at once. Kunjir et al. [4] found that combining data mining with visualization helps doctors see patterns and predict several diseases together. Vijiyarani and Sudha [5] reviewed many studies and concluded that classification and association-rule mining are very popular and effective for different kinds of diseases.

Nabeel et al. [6] reviewed hybrid models that mix machine learning with optimization techniques and found that these models give better prediction accuracy. Similarly, Kanakaraddi et al. [7] pointed out that scalable and distributed systems are important to make healthcare prediction practical on large datasets.

Kaur and Bawa [8] highlighted that big data tools are becoming more important in healthcare, especially as datasets keep growing. Chitra and Seenivasagam [9] argued that hybrid intelligent systems, like combining fuzzy logic, neural networks, and evolutionary algorithms, can give better results and are easier to interpret.

Other researchers focused specifically on heart disease prediction. David and Belcy [10] showed that standard data mining algorithms work well for symptom-based predictions. Ghani and Zainal [11] created a smart prediction system and found that good feature selection and model tuning are key to better outcomes.

Healthcare data in general has also been studied. Huang et al. [12] said that hospital information systems give lots of useful data for predictive models. Soni et al. [13] found

that machine learning models are usually more accurate than manual diagnoses when enough data is available. Dangare and Apte [14] improved heart disease predictions by adding extra clinical features and optimizing the classification models.

In short, previous research shows that good feature selection, scalable processing, and strong machine learning methods are all very important for disease prediction. However, most studies use medium-sized datasets and don't fully use big data tools. That's why this project focuses on using PySpark, MapReduce, and distributed computing to handle large healthcare datasets efficiently for disease prediction.

IV. Methodology and Implementation

A. Methodology Overview

For this project, we mainly wanted to make a machine learning pipeline that can predict diseases from a dataset with lots of symptoms. The dataset had many features, and we also wanted it to run fast with large data, so we decided to use Apache Spark. Spark is really good for this because it can process big data in parallel, manage memory well, and it comes with MLlib, which is a library for machine learning that works at a large scale.

We used PySpark so we could write Spark programs in Python. Spark SQL helped us load and clean the data, and MLlib let us build a Random Forest model. For things like plotting confusion matrices, checking classification reports, and looking at which features are important, we used normal Python libraries like pandas, matplotlib, seaborn, and scikit-learn.

The workflow we followed was simple, we first loaded the data, cleaned it, preprocessed it, combined all the features into vectors, trained the Random Forest model, tested how well it worked, and finally made some plots to better understand the results.

B. Data Collection and Preprocessing

The dataset comes as two CSV files, Training.csv and Testing.csv. Both files have columns for symptoms, where 0 means the symptom is not there and 1 means it is, and a column called prognosis, which tells us the disease. We loaded both files into Spark DataFrames to work with them.

Before we could train the model, we did some cleaning and preparation:

- Clean Column Names, some symptom names had spaces, hyphens, or other characters. We wrote a small function to clean them so the names matched in both files
- Type Conversion, all symptom columns were turned into integers so Spark ML could use them
- Label Encoding, the prognosis column was turned into numbers using StringIndexer. We fit it on the training data and applied it to the test data so the labels stayed consistent
- Vector Assembly, we combined all symptom columns into a single features vector using VectorAssembler

After this, the data was numeric, clean, and ready for Spark ML to process.

C. Implementation Steps

1) System Architecture: We built the system as a modular pipeline using Spark ML's Pipeline API. This made sure that all the steps, like cleaning, vectorizing, and modeling, worked the same way for both the training and test data. The main parts of the pipeline were:

1) StringIndexer to turn disease names into numbers
2) VectorAssembler to put all the symptoms into one features vector
3) RandomForestClassifier to predict the disease for each patient

This setup made it easier to repeat, adjust, and scale to bigger datasets.

2) Algorithms and Frameworks Used: We mainly used Spark MLlib tools:

- StringIndexer, to turn disease names into numbers
- VectorAssembler, to combine all symptom columns into one features column
- RandomForestClassifier, which works well with many features, is good with noisy data, and can handle multiple classes
- CrossValidator, to find the best model by testing different settings like number of trees and max depth
- MulticlassClassificationEvaluator, to check how good the model is using accuracy, precision, recall, and F1-score

3) Data Processing Workflow: Here's how we processed the data step by step:

1) Import Python and PySpark libraries
2) Load training and test datasets with Spark SQL
3) Clean all column names
4) Convert all feature columns to numbers
5) Fit StringIndexer on the training labels and apply it to the test set
6) Assemble all symptom columns into a features vector
7) Train the Random Forest model and tune it with cross-validation
8) Test the model on the test data
9) Convert predictions to Pandas to make plotting easier
10) Make confusion matrices, classification reports, and feature importance plots

4) Challenges and Solutions: We faced a few issues along the way:

- Messy Column Names, some symptom names had special characters which caused errors in Spark SQL, fixed with a cleaning function
- Unseen Labels, some diseases only appeared in the test data, solved by using a StringIndexer trained on the training data
- Too Many Features, with so many symptoms manually picking features was hard, so we automated feature selection
- Visualization in Spark, Spark cannot make plots easily, so we exported predictions to Pandas and used standard Python plotting libraries

D. Results and Discussion

The Random Forest model worked really well on the test dataset. Metrics like accuracy, precision, recall, and F1-score show that the model can predict most diseases correctly, see Table II. The confusion matrix also shows that most predictions fall on the diagonal, which means the model is classifying diseases the right way, see Table III.

TABLE II
RANDOM FOREST MODEL PERFORMANCE ON TEST DATA

Metric	Score
Accuracy	0.98
Weighted Precision	0.99
Weighted Recall	0.98
Weighted F1-Score	0.98

TABLE III
CONFUSION MATRIX OF RANDOM FOREST MODEL ON TEST DATA (SUBSET OF 10 DISEASES)

Actual / Predicted	Vertigo	AIDS	Acne	Allergy	Arthritis	Asthma	Cold	Dengue	Diabetes	GERD
Vertigo	1	0	0	0	0	0	0	0	0	0
AIDS	0	1	0	0	0	0	0	0	0	0
Acne	0	0	1	0	0	0	0	0	0	0
Allergy	0	0	0	1	0	0	0	0	0	0
Arthritis	0	0	0	0	1	0	0	0	0	0
Asthma	0	0	0	0	0	1	0	0	0	0
Cold	0	0	0	0	0	0	1	0	0	0
Dengue	0	0	0	0	0	0	0	1	0	0
Diabetes	0	0	0	0	0	0	0	0	1	0
GERD	0	0	0	0	0	0	0	0	0	1

We also checked which symptoms were the most important for the model's predictions. This helped us see which symptoms really mattered for telling diseases apart. From this, we can see that using Spark for distributed machine learning works well to find patterns in big healthcare data, and it could be useful for helping doctors predict diseases earlier and make better decisions.

V. SOURCE CODE

All the code for this project is included in the Jupyter Notebook we submitted. The notebook is organized into clear sections and has comments that explain what each part does, so it's easy to follow and reproduce. It covers the full workflow from start to finish: loading the data with Spark SQL, cleaning column names, converting features to numeric types, encoding disease labels with StringIndexer, combining features using VectorAssembler, and training the Random Forest model with cross-validation.

The notebook also contains all the analysis and visualizations that support our findings, like confusion matrices, classification reports, and feature importance plots. Anyone reading the notebook can see exactly how each step of the methodology was implemented and can reproduce the results shown in this report.

VI. PRESENTATION

We gave an 8-minute presentation to show what we did in this project. The goal was to explain our system, how we built it, and what we found, in a way that's easy to follow. The main points we talked about were:

- Project Goals: We explained that the project is about predicting diseases from patient symptoms using Apache Spark. We focused on handling a lot of symptom data and building a model that can classify multiple diseases at once.
- How We Did It: We walked through the steps we used: cleaning column names, converting features to numbers, turning the disease names into labels with StringIndexer, combining features with VectorAssembler, and training a RandomForestClassifier. We also showed how we used Spark DataFrames, Pipelines, and cross-validation to make the model work better.
- Results: We showed how well our model worked using accuracy, precision, recall, and F1-score. We also showed a confusion matrix and a feature importance chart to see which symptoms mattered most. These plots were made in Python after we converted our Spark predictions to Pandas, and they helped explain how the model makes decisions.
- Challenges and Future Ideas: We talked about some tricky parts, like dealing with lots of features, diseases that have similar symptoms, and some labels that didn't have many examples. For future work, we suggested using more data, trying other models like Gradient Boosted Trees or Neural Networks, doing feature selection, and maybe making a real-time system for hospitals.

Overall, we tried to take the audience step by step through what we did and why. Using charts and summaries at each stage helped make the workflow and results easy to understand.

VII. CONCLUSION

This project successfully demonstrated the application of big data analytics and distributed machine learning for disease prognosis prediction using PySpark. By preprocessing a large symptom-based dataset, performing feature engineering, and training a Random Forest model, we were able to achieve high predictive performance across multiple disease categories. The results, supported by evaluation metrics and confusion matrix analysis, highlight the potential of scalable frameworks like Spark to handle high-dimensional healthcare data efficiently.

Overall, the project provided valuable hands-on experience in end-to-end data processing, model development, and interpretation of predictive insights. It reinforced the importance of clean data, proper feature handling, and rigorous evaluation, while showing how distributed machine learning can bridge the gap between raw medical data and actionable healthcare insights.

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to Mr. Umid Suleymanov, the instructor of INFT-4836 at ADA University, for his valuable guidance and constructive feedback throughout this project.

REFERENCES

[1] N. Bhatla and K. Jyoti, "An analysis of heart disease prediction using different data mining techniques," in *Proc. Int. Conf. Contemporary Computing*, 2012.

[2] A. K. Bhatnagar, P. Madan, A. Rana, S. Sharma, S. Sonawane, and C. V. Josphine, "An efficient techniques for disease prediction from medical data using data mining and machine learning," in *Proc. 5th Int. Conf. Contemporary Computing and Informatics (IC3I)*, 2022, pp. 839–845.

[3] K. Srinivas, B. K. Rani, and A. Govrdhan, "Applications of data mining techniques in healthcare and prediction of heart attacks," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 2, pp. 250–255, 2010.

[4] A. Kunjir, H. Sawant, and N. F. Shaikh, "Data mining and visualization for prediction of multiple diseases in healthcare," Modern Education Society's College of Engineering, Pune, India.

[5] S. Vijiyarani and S. Sudha, "Disease prediction in data mining technique – A survey," *Dept. of Computer Science, Bharathiar University*, India.

[6] M. Nabeel *et al.*, "Review on effective disease prediction through data mining techniques," *Int. J. Electr. Eng. Informatics*, vol. 13, no. 3, pp. 540–552, 2021, doi:10.15676/ijeei.2021.13.3.13.

[7] S. G. Kanakaraddi, K. C. Gull, J. Bali, A. K. Chikaraddi, and S. Giraddi, "Disease prediction using data mining and machine learning techniques," in *Advanced Prognostic Predictive Modelling in Healthcare Data Analytics*. Springer, 2021, pp. 71–92.

[8] S. Kaur and R. K. Bawa, "Future trends of data mining in predicting the various diseases in medical healthcare system," Punjabi University, Patiala, India.

[9] R. Chitra and V. Seenivasagam, "Review of heart disease prediction system using data mining and hybrid intelligent techniques," *Dept. of CSE / IT*, India.

[10] H. B. F. David and S. A. Belcy, "Heart disease prediction using data mining techniques," Manonmaniam Sundaranar Univ., India.

[11] M. F. M. Ghani and A. A. A. Zainal, "Intelligent heart disease prediction system using data mining techniques," in *Proc. Int. Conf. Advanced Information Networking and Applications Workshops*, 2008, doi:10.1109/AICCSA.2008.4493524.

[12] F. Huang, S. Wang, and C.-C. Chan, "Predicting disease by using data mining based on healthcare information system," Univ. of Akron, OH, USA.

[13] J. Soni, U. Ansari, D. Sharma, and S. Soni, "Predictive data mining for medical diagnosis: An overview of heart disease prediction," Raipur Institute of Technology and Bhilai Institute of Technology, India.

[14] C. S. Dangare and S. S. Apte, "Improved study of heart disease prediction system using data mining classification techniques," Walchand Institute of Technology, Solapur, India.