A Comparative Analysis of Diabetes Classification through Diverse Machine Learning Approaches

A MINI PROJECT

Submitted by

Riyan Acharya(RA2111027010084)

Under the guidance

of **Dr. E. Sasikala**

Professor

Department of Data Science and Business Systems

In partial fulfillment for

the Course of

18CSE392T- Machine Learning-I

in

Department of Data Science and Business Systems

SCHOOL OF COMPUTING

COLLEGE OF ENGINEERING AND TECHNOLOGY
SRM INSTITUTE OF SCIENCE AND TECHNOLOGY KATTANKULATHUR – 603203
October 2023

COLLEGE OF ENGINEERING & TECHNOLOGY
SRM INSTITUTE OF SCIENCE & TECHNOLOGY
S.R.M. NAGAR, KATTANKULATHUR – 603 203

BONAFIDE CERTIFICATE

Certified that this mini project report "A Comparative Analysis of Diabetes Classification through Diverse Machine Learning Approaches" is the Bonafide work of **Riyan Acharya(RA2111027010084)** who carried out the project work under my supervision.

Dr. E. Sasikala
Professor
Department of Data Science and Business Systems
SRM institute of science and technology

Dr. M Lakshmi
Professor & HOD
Department of DSBS
SRM institute of science and technology

ABSTRACT

Diabetes is a chronic metabolic disorder that affects millions of people worldwide. Early detection and effective management of diabetes are crucial to prevent severe complications and improve the quality of life for affected individuals. Symptoms-based diabetes detection is a beneficial approach for addressing the disease at its early stage. Machine Learning (ML) techniques have shown great promise in aiding the early detection of various diseases on patient data, including diabetes. ML algorithms can analyze vast datasets, identify patterns, and make accurate predictions, helping medical professionals to diagnose diabetes at its early stages. In our work, we employed several ML models for diabetes classification using different datasets. These models include K- Nearest Neighbour (KNN), Logistic Regression (LR), Decision Tree (DT), Random Forest (RF), Gradient-Boosted Model (GBM), eXtreme Gradient Boosting (XG-Boost), Adaptive Boosting (Ada-Boost), Support Vector Machine (SVM), and Gaussian Naive Bayes (GNB). We performed a comparative analysis of their performance on three distinct datasets using evaluation metrics like accuracy, precision, F1-score, sensitivity, specificity, and Cohen's Kappa Value. Our findings revealed that the RF algorithm is optimal for symptoms-based and primary lab report-based diabetes detection, while XG-Boost excels in classifying different types of diabetes from a multi-class dataset. Moreover, we investigated diverse symptoms and their impact on diabetes outcomes, offering insights into preventive measures and early-stage monitoring for this disease classification.

TABLE OF CONTENTS

OBJECTIVE

Aim:

To determine diabetes from three different datasets and show a comparative analysis between the different algorithms

Synopsis:

Diabetes is a harmful chronic disease that impacts millions of people globally, causing various complications such as disabilities, infections, and, in some cases, death. It is a chronic metabolic disorder characterized by high blood sugar levels due to either insufficient insulin production by the body or improper insulin utilization by cells.

Due to globalization, modernization, and urbanization, diabetes is more common in developing countries, resulting in reduced physical activity and altered eating habits. For instance, in India, urban areas exhibit a prevalence rate of 20%, slightly surpassing the 10% rate in rural regions. To mitigate the risk of diabetes, adopting a healthy lifestyle and proper dietary choices are essential. Early and accurate detection of diabetes is crucial for achieving this goal.

In this situation, machine learning (ML) is essential for detecting diabetes by analyzing symptoms, everyday lifestyle and activities, and other contributing factors. Recently, ML has emerged as a powerful tool in diabetes detec- tion, aiding healthcare professionals in early diagnosis and personalized treatment planning. ML models can detect patterns indicative of diabetes at an early stage, enabling timely intervention and improved treatment outcomes. Moreover, ML algorithms can analyze individual patient data to recommend personalized treatment plans based on their specific health profiles. For diabetes classification, datasets consisting of various patient characteristics, such as age, gender, family history, lifestyle habits, blood pressure, and blood glucose levels, are collected from electronic health records, medical devices, and surveys, and the data is preprocessed to handle missing values, normalize features, and remove outliers. It ensures that the data is suitable for ML model training. Various ML models are already utilized for the purpose of diabetes classification. The aim of this paper is to enhance this approach through a comparative study on symptoms-based classification, as well as different types of diabetes detection.

In this work, we employed various ML algorithms for diabetes classification, including KNN, LR, DT , RF , GBM , XG-Boost , Ada-Boost , SVM , and GNB . These algorithms were applied to three distinct datasets: one based on external symptoms, another based on primary lab test reports, and a third one for different varieties of diabetes classification, including diabetes type-1 and type-2. We conducted a comparative analysis for the efficiency of the models using multiple evaluation metrics, such as Accuracy, Precision, F1-Score, Sensitivity, and Specificity on the three datasets separately. We finally determined the best algorithm for each dataset.

Among the nine ML models tested, the RF algorithm exhibited the best performance for diabetes detection based on both symptoms and lab reports, while XG-Boost excelled at classifying different types of diabetes. RF classifier algorithm showed 96.15% and 91.42% accuracy for symptoms-based and lab report-based diabetes detection datasets, respectively, which is the highest among all the utilized ML models. The XG-Boost model showed 96.1% accuracy in classifying the different types of diabetes diseases. Additionally, We examined various symptoms and assessed their effects on diabetes

classification, proposing potential preventive measures for detecting this fatal disease.

REQUIREMENT SPECIFICATIONS

HARDWARE AND SOFTWARE SPECIFICATION

HARDWARE REQUIREMENTS

- Hard disk: 512 GB and above.
- Processor: ryzen 5 or equivalent and above.
- Ram: 8GB and above.

SOFTWARE REQUIREMENTS

- Operating System: Windows 10
- Software: python
- Tools: Google Collab

TECHNOLOGIES USED

- Programming Language: **Python**

INTRODUCTION TO PYTHON

Python is a widely used general-purpose, high level programming language. It was initially designed by Guido van Rossum in 1991 and developed by Python Software Foundation. It was mainly developed for emphasis on code readability, and its syntax allows programmers to express concepts in fewer lines of code. Python is a programming language that lets you work quickly and integrate systems more efficiently.

It is used for:

- web development (server-side),
- software development,
- mathematics,
- System scripting.

What can Python do?

- Python can be used on a server to create web applications.
- Python can be used alongside software to create workflows.
- Python can connect to database systems. It can also read and modify files.
- Python can be used to handle big data and perform complex mathematics.
- Python can be used for rapid prototyping, or for production-ready software development.

Why Python?

- Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc.).
- Python has a simple syntax like the English language.
- Python has syntax that allows developers to write programs with fewer lines than some other programming languages.
- Python runs on an interpreter system, meaning that code can be executed as soon as it is written. This means that prototyping can be very quick.
- Python can be treated in a procedural way, an object-orientated way, or a functional way.

Good to know.

- The most recent major version of Python is Python 3, which we shall be using in this tutorial. However, Python 2, although not being updated with anything other than security updates, is still quite popular.
- Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been backported to Python 2. But in general, they remain not quite compatible.
- Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official End of Life date of January 1, 2020, has been established for Python 2, after which time it will no longer be maintained.
- Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator for Life) by the 12 Python community. The name Python derives not from the snake, but from the British comedy troupe Monty Python's Flying Circus, of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.
- It is possible to write Python in an Integrated Development Environment, such as Thonny, PyCharm, NetBeans or Eclipse which are particularly useful when managing larger collections of Python files.

Python Syntax compared to other programming languages.

- Python was designed to for readability and has some similarities to the English language with influence from mathematics.
- Python uses new lines to complete a command, as opposed to other programming languages which often use semicolons or parentheses.
- Python relies on indentation, using whitespace, to define scope, such as the scope of loops, functions, and classes. Other programming languages often use curly brackets for this purpose.

Python is Interpreted

- Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer's processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.
- This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.
- One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.
- In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The 13 expediency of coding in an interpreted language is typically worth it for most applications.
- For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and object-oriented programming.
- Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.
- Python accomplishes what many programming languages don't: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

Machine learning

Introduction:

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without using explicit instructions, relying on patterns and inference instead. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to perform the task. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop a conventional algorithm for effectively performing the task.

Machine learning is closely related to computational statistics, which focuses on making predictions using computers. The study of mathematical optimization delivers methods, theory and application domains to the field of machine learning. Data mining is a field of study within machine learning, and focuses on exploratory data analysis through learning. In its application across business problems, machine learning is also referred to as predictive analytics.

Machine learning tasks:

Machine learning tasks are classified into several broad categories. In supervised learning, the algorithm builds a mathematical model from a set of data that contains both the inputs and the desired outputs. For example, if the task were determining whether an image contained a certain object, the training data for a supervised learning algorithm would include images with and without that object (the input), and each image would have a label (the output) designating whether it contained the object. In special cases, the input may be only partially available, or restricted to special feedback. Semi-algorithms develop mathematical models from incomplete training data, where some sample input doesn't have labels. Classification algorithms and regression algorithms are types of supervised

learning. Classification algorithms are used when the outputs are restricted to a limited set

of values. For a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email. For an algorithm that identifies spam emails, the output would be the prediction of either "spam" or "not spam", represented by the Boolean values true and false. Regression algorithms are named for their continuous outputs, meaning they may have any value within a range. Examples of a continuous value are the temperature, length, or price of an object. In unsupervised learning, the algorithm builds a mathematical model from a set of data that contains only inputs and no desired output labels. Unsupervised learning algorithms are used to find structure in the data, like grouping or clustering of data points. Unsupervised learning can Discover patterns in the data, and can group the inputs into categories, as in feature learning. Dimensionality reduction is the process of reducing the number of "features", or inputs, in a set of data. Active learning algorithms access the desired outputs (training labels) for a limited set of inputs based on a budget and optimize the choice of inputs for which it will acquire training labels. When used interactively, these can be presented to a human user for labeling. Reinforcement learning algorithms are given feedback in the form of positive or negative reinforcement in a dynamic environment and are used in autonomous vehicles or in learning to play a game against a human opponent. Other specialized algorithms in machine learning include topic modeling, where the computer program is given a set of natural language documents and finds other documents that cover similar topics. Machine learning algorithms can be used to find the unobservable probability density function in density estimation problems. Meta learning algorithms learn their own inductive bias based on previous experience. In developmental robotics, robot learning algorithms generate their own sequences of learning experiences, also known as a curriculum, to cumulatively acquire new skills through self-guided exploration and social interaction with humans. These robots use guidance mechanisms such as active learning, maturation, motor synergies, and imitation.

Types of learning algorithms:

The types of machine learning algorithms differ in their approach, the type of data they input and output, and the type of task or problem that they are intended to solve.

Supervised learning:
Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data and consists of a set of training examples. Each training example has one or more inputs and the desired output, also known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task. Supervised learning algorithms include classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification. In the case of semi-supervised learning algorithms, some of the training examples are missing training labels, but they can nevertheless be used to improve the quality of a model. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however, these labels are often cheaper to obtain, resulting in larger effective training sets.

Unsupervised learning:
Unsupervised learning algorithms take a set of data that contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified, or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, though unsupervised learning encompasses other domains involving summarizing and explaining data features. Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric, and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

Semi-supervised learning:
Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy.

K-Nearest Neighbors
Introduction In four years of analytics built more than 80% of classification models and just 15- 20% regression models. These ratios can be generalized throughout the industry. The reason for a bias towards classification models is that most analytical problems involve making a decision. For instance, will a customer attrite or not, should we target customer X for digital campaigns, whether customer has a high potential or not etc. This analysis is more insightful and directly links to an implementation roadmap. In this article, we will talk about another widely used classification technique called K-nearest neighbors (KNN). Our focus will be primarily on how does the algorithm work and how does the input parameter effect the output/prediction.

KNN algorithm
KNN can be used for both classification and regression predictive problems. However, it is more widely used in classification problems in the industry. To evaluate any technique, we generally look at 3 important aspects:
1. Ease to interpret output
2. Calculation time
3. Predictive Power

Decision tree
In a decision tree, the algorithm starts with a root node of a tree then compares the value of different attributes and follows the next branch until it reaches the end leaf node. It uses different algorithms to check the split and variable that allow the best homogeneous sets of population. decision trees are widely used in data science. It is a key proven tool for making decisions in complex scenarios. In Machine learning, ensemble methods like decision tree, random forest are widely used. Decision trees are a type of supervised learning algorithm where data will continuously be divided into different categories according to certain parameters. So, in this blog, I will explain the Decision tree algorithm. How is it used? How its functions will cover everything that is related to the decision tree.

What is a Decision Tree?

Decision tree as the name suggests is a flow-like a tree structure that works on the principle of conditions. It is efficient and has strong algorithms used for predictive analysis. It has mainly been attributed to internal nodes, branches, and a terminal node. Every internal node holds a “test” on an attribute, branches hold the conclusion of the test, and every leaf node means the class label. This is the most used algorithm when it comes to supervised learning techniques. It is used for both classifications as well as regression. It is often termed as “CART” that means Classification and Regression Tree. Tree algorithms are always preferred due to stability and reliability.

How can an algorithm be used to represent a tree Let us see an example of a basic decision tree where it is to be decided in what conditions to play cricket and in what conditions not to play. You might have got a fair idea about the conditions on which decision trees work with the above example. Let us now see the common terms used in Decision Tree that is stated below:

- Branches - The division of the whole tree is called branches.
- Root Node - Represent the whole sample that is further divided.
- Splitting - The division of nodes is called splitting.
- Terminal Node - Node that does not split further is called a terminal node.
- Decision Node - It is a node that also gets further divided into different sub-nodes being a sub node.
- Pruning - Removal of sub-nodes from a decision node.
- Parent and Child Node - When a node gets divided further then that node is termed as parent node whereas the divided nodes or the sub-nodes are termed as a child node of the parent node.

DATASET DESCRIPTION

We used three different datasets collected from Kaggle, for our comparative analysis. We labeled three datasets for our convenience: dataset-1 for symptoms-based diabetes detection, dataset-2 for primary lab test report-based diabetes detection, and dataset-3 for different types of diabetes classification.

Dataset-1 includes 16 symptoms and corresponding diabetes outcomes from 520 male and female patients. The symptoms encompass age, gender, weakness, itching, obesity, and more.

Dataset-2 comprises some primary lab test reports, such as glucose level, insulin, body mass index (BMI), blood pressure, pregnancy days, etc, to name a few, with the corresponding diabetes outcomes. It includes 768 patients' data, considering both male and female patients.

Dataset 3 covers 21 different conditions and their possible outcomes. It is a multi-class classification dataset, where the outcomes indicate non-diabetes, diabetes type-1, or diabetes type-2. The conditions include food habits, past disease history, physical and mental health, etc. The dataset comprises 253,680 patients of different genders, which is huge in size.

Data Pre-Processing

Data pre-processing is a crucial step in any data analysis or ML project. It involves cleaning, transforming, and organizing raw data to make it suitable for further analysis and modeling. The steps involving data pre-processing for our work have been discussed here.

Data loading:

The first step in data pre-processing is to read the CSV data into a suitable data structure. Python provides various libraries like pandas and numpy to handle CSV files effectively.

Handling Missing Data:

Next, dealing with missing data is critical as it can adversely affect the accuracy and reliability of your analysis.

We handled the missing data by filling in missing values with means of the remaining data.

Encoding Categorical Variables:

Then, categorical variables need to be converted into numerical representations. As example, for gender column of the dataset, we labeled the female and male as 0 and 1 respectively, to convert categorical data into binary form.

Splitting the Data:

Then the dataset is split into training and testing sets. The model is trained on the training set and its performance is evaluated on the testing set, to assess how well the model generalizes to new, unseen data. In our experiment, we trained using 90% of the data and tested with the remaining 10% for each dataset, individually.

Results and Discussions

Evaluation Metrics

Evaluation metrics are invaluable tools for quantifying performance and effectiveness in various domains, including ML. They provide quantifiable measures that help to determine how well a specific task has been accomplished. In this work, we used different evaluation metrics, such as accuracy, precision, recall, F1-score, sensitivity, specificity, and Cohen's Kappa value.

For a classification task, accuracy signifies the proportion of correctly classified instances out of the total samples. The ratio of true positive predictions to the total positive predictions, measuring the model's ability to avoid false positives, is termed as precision. Recall refers to the ratio of true positive predictions to the total actual positives, measuring the model's ability to find all positive instances. The harmonic mean of precision and recall is called F1-Score, providing a balanced evaluation metric when there is an imbalance between positive and negative classes. Sensitivity and specificity are vital measures for understanding the performance of diagnostic tests and prediction models. Sensitivity measures the proportion of true positive results among all individuals who actually have the condition being tested for, while specificity indicates the proportion of true negative results among all individuals who do not have the condition being examined.

Cohen's Kappa is a statistical measure that assesses the level of agreement between two raters beyond what would be expected by chance alone. It helps researchers and practitioners determine the consistency of ratings and the quality of data collected by multiple raters.

Comparative Analysis

The nine ML models (KNN, LR, DT, RF, GBM, XGBoost, Ada-Boost, SVM and GNB) are separately trained and tested using the three datasets. The comparison of all the models for each dataset is presented individually, and the best model for each dataset is discovered from the comparative analysis.

Table 1 presents accuracy (%), precision, recall, F1-score, sensitivity, specificity, and Cohen's Kappa values for all ML models on dataset-1. The RF model stands out with an impressive accuracy of 96.15%, indicating superior performance. Notably, XGBoost, LR, and GNB models also achieved over 90% accuracy, significantly outperforming other models. Fig. 1 displays correlations between features in Dataset-1 for potential diabetes outcomes. The features represent various diabetes-associated symptoms present in Dataset-1.

Models	Accuracy	Precision	Recall	F1 Score	Sensitivity	Specificity	Cohen's Kappa Value
KNN	82.69	0.79	0.76	0.78	0.82	0.83	0.632
LR	88.46	0.92	0.92	0.92	0.91	0.83	0.745
DT	78.84	0.64	0.89	0.74	0.73	0.88	0.571
RF	96.15	0.92	0.96	0.94	0.97	0.94	0.915
GBM	65.38	0.65	0.62	0.64	1	0	0
XGBoost	90.38	0.88	0.83	0.86	0.94	0.83	0.784
Ada-Boost	78.84	0.64	0.79	0.84	0.73	0.88	0.571
SVM	65.38	0.65	0.71	0.69	1	0	0.554
GNB	92.3	0.94	0.83	0.88	0.97	0.83	0.825

Table 1: Performance evaluation metrics corresponding to each individual ML models on Dataset-1 (Symptoms-based Diabetes Dataset)

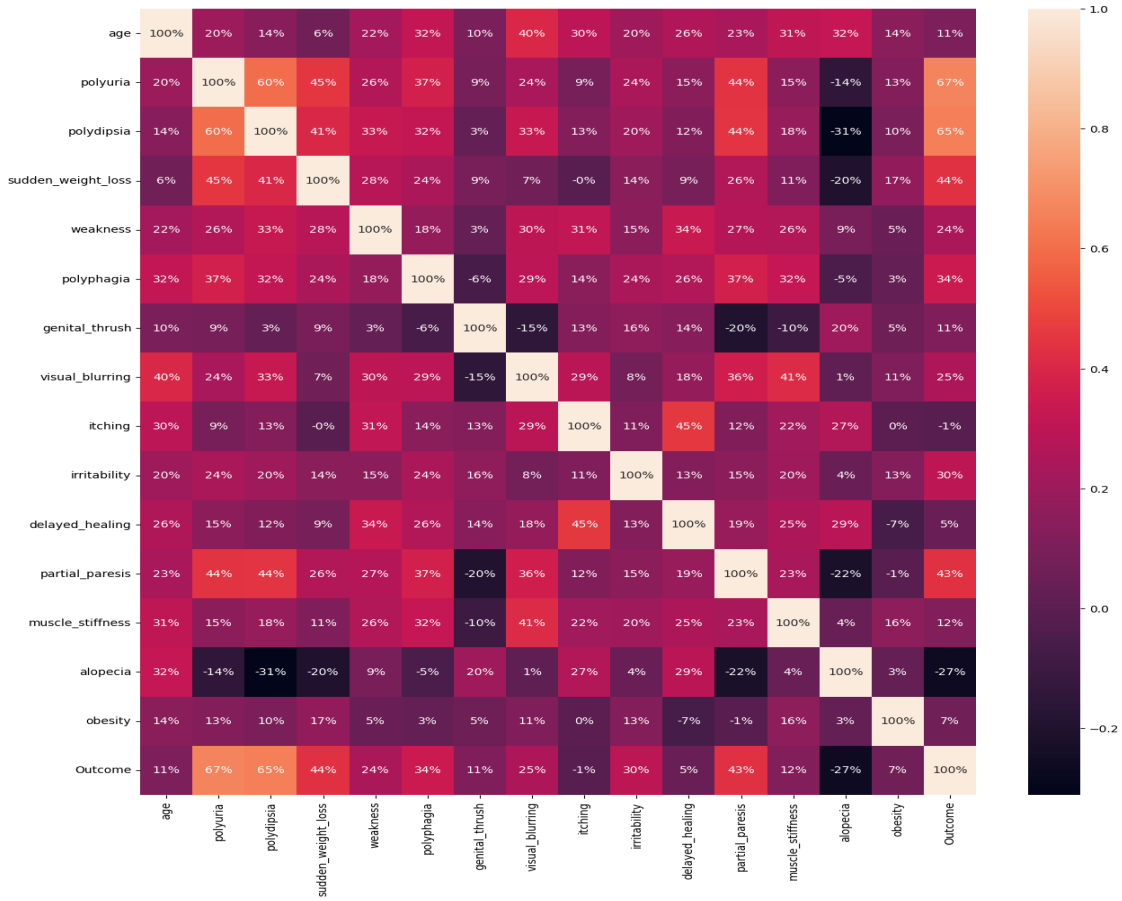

Fig. 1: A Diagrammatic Representation of Correlations between the Features in Dataset-1 for Possible Diabetes Outcomes.

Table 2 presents the values of the same evaluation metrics for all ML models on dataset-2. Here also The RF model achieved the highest accuracy (91.42%), showing the best performance. On this dataset, SVM also showed significantly more impressive accuracy than the rest of the models. Fig. 2 shows the correlations between the features in Dataset-2 for potential diabetes outcomes, with each feature linked to the primary lab reports.

Models	Accuracy	Precision	Recall	F1 Score	Sensitivity	Specificity	Cohen's Kappa Value
KNN	74.57	0.78	0.84	0.81	0.55	0.84	0.409
LR	79.22	0.81	0.88	0.85	0.62	0.88	0.482
DT	81.81	0.88	0.84	0.86	0.77	0.84	0.234
RF	91.42	0.87	0.8	0.87	0.55	0.8	0.474
GBM	64.93	0.65	1	0.79	0	1	0
XGBoost	83.11	0.84	0.92	0.88	0.33	0.86	0.234
Ada-Boost	76.62	0.75	0.96	0.84	0.4	0.96	0.235
SVM	87.92	0.88	0.92	0.84	0.51	0.88	0.323
GNB	79.22	0.75	0.86	0.8	0.6	0.86	0.482

Table 2: Performance evaluation metrics corresponding to each individual ML models for Dataset-2 (Primary Lab Report based Diabetes Dataset)

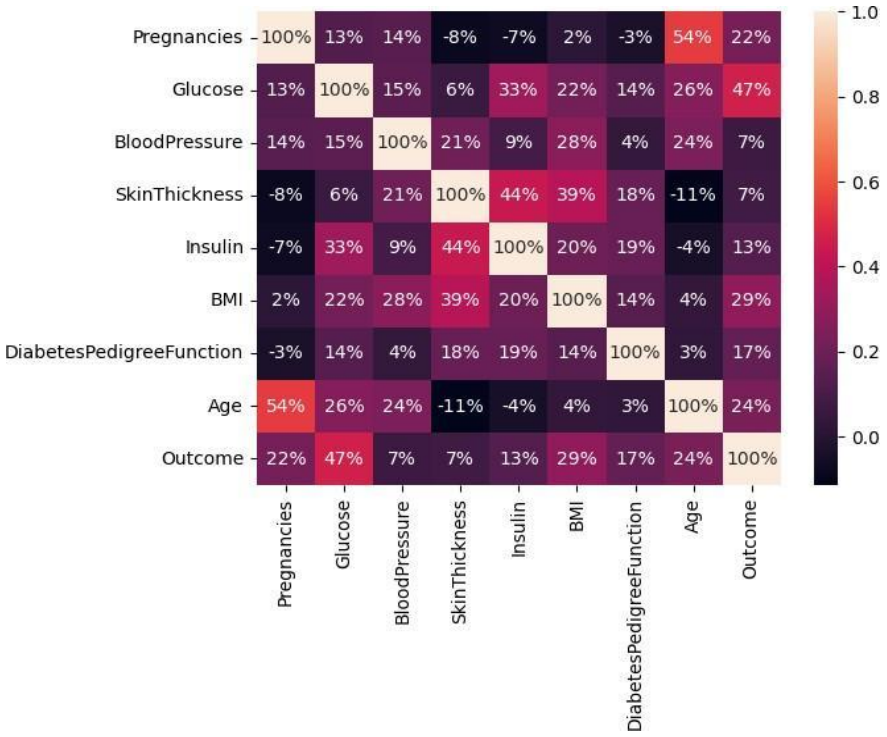

Fig. 2: A Diagrammatic Representation of Correlations between the Features in Dataset-2 for Possible Diabetes Outcomes

Table 3 indicates the comparative performance of all ML models on dataset-3. Here, XGBoost outperformed all other models for classifying different types of diabetes, with an accuracy of 96.1%. The performance of RF is closer to the XGBoost, however, the accuracy of other models is comparatively lower than these two models. Fig. 3 presents the correlations between the features in dataset 3. These features indicate different symptoms and their possible outcomes for different types of diabetes classification.

Models	Accuracy	Precision	Recall	F1 Score	Sensitivity	Specificity	Cohen's Kappa Value
KNN	84.9	0.86	0.98	0.92	0.84	0.86	0.185
LR	85.2	0.87	0.98	0.92	0	1	0.175
DT	84.1	0.84	1	0.91	0	1	0.198
RF	90.1	0.9	0.94	0.91	0.0054	0.997	0.197
GBM	84.1	0.85	1	0.9	0	1	0
XGBoost	96.1	0.96	0.95	0.98	0.78	0.94	0.278
Ada-Boost	84.1	0.84	1	0.91	0	1	0
SVM	83.8	0.84	0.98	0.91	0.055	0.965	0.187
GNB	75.8	0.91	0.81	0.85	0.0426	0.9891	0.279

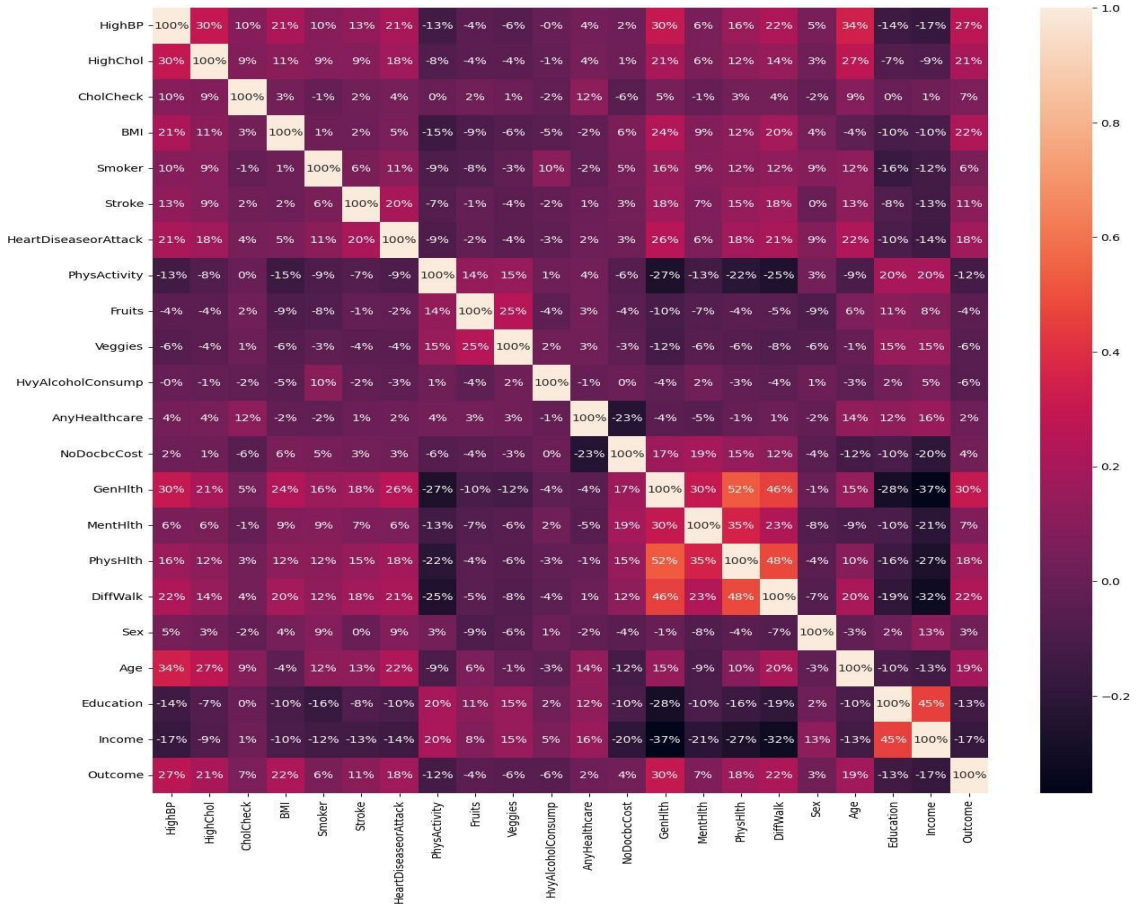

Fig. 3: A Diagrammatic Representation of Correlations between the Features in Dataset-3 for Possible Diabetes Outcomes.

After conducting a comparative study, we concluded that the RF algorithm is the most suitable for symptoms-based and primary lab report-based diabetes detection datasets. On the other hand, XG-Boost exhibited the best performance for the different types of diabetes classification tasks using the multi-class diabetes dataset. Nonetheless, the RF model's performance in different types of diabetes classification was also highly competitive compared to other models. Additionally, we showed the correlations between the features for all three datasets. This correlation signifies the features are related to each other, as well as the impact of each feature on the outcomes. For instance, Fig. 2 suggests that glucose and BMI significantly influence diabetes outcomes compared to other factors. Similarly, Fig. 1 indicates that polyuria and polydipsia play a crucial role in diabetes, while Fig. 3 suggests that general health may also strongly affect the possibility of diabetes outcomes.

Conclusion and Future Works

In this paper, we presented a comparative study of various ML algorithms for classifying diabetes based on different symptoms. Recently, ML has demonstrated tremendous potential in diabetes detection and management. However, with the significant advancement of ML for this purpose, there remains ample scope for further research and development in this domain. As research in this field progresses, integrating big data, adopting explainable AI techniques, promoting personalized medicine, ensuring data privacy, and facilitating real-time monitoring will pave the way for more accurate, accessible, and patient-centric diabetes care. By combining the strengths of technology and medical expertise, we can collectively work towards a future where diabetes is detected early, and managed effectively, and its burden on individuals and healthcare systems is significantly reduced.

DATASET LINK: https://drive.google.com/drive/folders/1POaeFTZ9zliOdvkv9I7baaoBK_OGu5ui? usp=sharing

Code Links:
https://colab.research.google.com/drive/1_h9q6fGiBIsxsa9H5T_OM95pnYvs5NVY?usp=sharing
https://colab.research.google.com/drive/1Cm3s2C5CarfqdNa58_HWy8_8m5b3vqp3?usp=sharing
https://colab.research.google.com/drive/1sdi4g65JoS2Wsm7lvSUc9JxPx2u1AkS5?usp=sharing

REFERENCES

[1] Abdulhadi, N., Al-Mousa, A.: Diabetes detection using machine learning classification methods. In: 2021 International Conference on Infor- mation Technology (ICIT). pp. 350–354. IEEE (2021)
[2] Atkinson, M.A., Eisenbarth, G.S., Michels, A.W.: Type 1 diabetes. The Lancet **383**(9911), 69–82 (2014)
[3] Chatterjee, S., Khunti, K., Davies, M.J.: Type 2 diabetes. The lancet **389**(10085), 2239–2251 (2017)
[4] Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016)
[5] Cristianini, N., Ricci, E.: Support Vector Machines, pp. 928–932. Springer US, Boston, MA (2008)

[1] Dalianis, H., Dalianis, H.: Evaluation metrics and evaluation. Clinical Text Mining: secondary use of electronic patient records pp. 45–53 (2018)
[2] Deshpande, A.D., Harris-Hayes, M., Schootman, M.: Epidemiology of diabetes and diabetes-related complications. Physical therapy **88**(11), 1254–1264 (2008)
[3] DiMeglio, L.A., Evans-Molina, C., Oram, R.A.: Type 1 diabetes. The Lancet **391**(10138), 2449–2462 (2018)
[4] Gahukar, G., Gahukar, G.: Classification algorithms in machine learning (2019)
[5] Garc´ıa, S., Ram´ırez-Gallego, S., Luengo, J., Ben´ıtez, J.M., Herrera, F.: Big data preprocessing: methods and prospects. Big Data Analytics
1(1), 1–22 (2016)
[6] Ghosh, P., Azam, S., Karim, A., Hassan, M., Roy, K., Jonkman, M.: A comparative study of different machine learning tools in detecting diabetes. Procedia Computer Science **192**, 467–477 (2021)
[7] Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: European conference on information retrieval. pp. 345–359. Springer (2005)
[8] Goutte, C., Gaussier, E.: A probabilistic interpretation of precision, recall and f-score, with implication for evaluation. In: European conference on information retrieval. pp. 345–359. Springer (2005)
[9] Gujral, S.: Early diabetes detection using machine learning: a review (2017)
[10] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: Knn model-based approach in classification. In: On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings. pp. 986–996. Springer (2003)
[11] Gupta, O., Joshi, M., Dave, S.: Prevalence of diabetes in india. Advances in metabolic disorders **9**, 147–165 (1978)
[12] He, B., Shu, K.i., Zhang, H.: Machine learning and data mining in diabetes diagnosis and treatment. In: IOP conference series: materials science and engineering. vol. 490, p. 042049. IOP Publishing (2019)
[13] Jahromi, A.H., Taheri, M.: A non-parametric mixture of gaussian naive bayes classifiers based on local independent features. In: 2017 Artificial intelligence and signal processing conference (AISP). pp. 209–212. IEEE (2017)
[14] Lin, C.H., Chang, Y.C., Chuang, L.M.: Early detection of diabetic kidney disease: Present limitations and future perspectives. World journal of diabetes **7**(14), 290 (2016)
[15] Liu, Y., Wang, Y., Zhang, J.: New machine learning algorithm: Random forest. In: Liu, B., Ma, M., Chang, J. (eds.) Information Computing and Applications. pp. 246–252. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
[16] McKinney, W., et al.: pandas: a foundational python library for data analysis and statistics. Python for high performance and scientific com- puting **14**(9), 1–9 (2011)
[17] Mesquita, F., Maur´ıcio, J., Marques, G.: Oversampling techniques for diabetes classification: A comparative study. In: 2021 International Conference on e-Health and Bioengineering (EHB). pp. 1–6. IEEE (2021)
[18] Misra, A., Gopalan, H., Jayawardena, R., Hills, A.P., Soares, M., Reza-Albarra´n, A.A., Ramaiya, K.L.: Diabetes in developing countries. Journal of diabetes **11**(7), 522–539 (2019)
[19] Mujumdar, A., Vaidehi, V.: Diabetes prediction using machine learning algorithms. Procedia Computer Science **165**, 292–299 (2019)
[20] Nick, T.G., Campbell, K.M.: Logistic regression. Topics in biostatistics pp. 273–301 (2007)
[21] Oliphant, T.E., et al.: Guide to numpy, vol. 1. Trelgol Publishing USA (2006)
[22] Papatheodorou, K., Banach, M., Bekiari, E., Rizzo, M., Edmonds, M., et al.: Complications of diabetes 2017 (2018)
[23] Quinlan, J.R.: Learning decision tree classifiers. ACM Computing Surveys (CSUR) **28**(1), 71–72 (1996)

[24] Rady, M., Moussa, K., Mostafa, M., Elbasry, A., Ezzat, Z., Medhat, W.: Diabetes prediction using machine learning: A comparative study. In: 2021 3rd Novel Intelligent and Leading Emerging Sciences Conference (NILES). pp. 279–282 (2021). https://doi.org/10.1109/NILES53778.2021.9600091
[25] Ramachandran, A., Snehalatha, C.: Current scenario of diabetes in india. Journal of diabetes **1**(1), 18–28 (2009)
[26] Roglic, G., et al.: Who global report on diabetes: A summary. International Journal of Noncommunicable Diseases **1**(1), 3 (2016)
[27] Sankar Ganesh, P., Sripriya, P.: A comparative review of prediction methods for pima indians diabetes dataset. Computational Vision and Bio-Inspired Computing: ICCVBIC 2019 pp. 735–750 (2020)
[28] Schapire, R.E.: Explaining adaboost. In: Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik, pp. 37–52. Springer (2013)
[29] Swapna, G., Vinayakumar, R., Soman, K.: Diabetes detection using deep learning algorithms. ICT express **4**(4), 243–246 (2018)
[30] Swapna, G., Vinayakumar, R., Soman, K.: Diabetes detection using deep learning algorithms. ICT express **4**(4), 243–246 (2018)
[31] Swift, A., Heale, R., Twycross, A.: What are sensitivity and specificity? Evidence-Based Nursing **23**(1), 2–4 (2020)
[32] Vach, W.: The dependence of cohen's kappa on the prevalence does not matter. Journal of clinical epidemiology **58**(7), 655–661 (2005)
[33] Vijan, S.: Type 2 diabetes. Annals of internal medicine **152**(5), ITC3–1 (2010)
[34] Yacouby, R., Axman, D.: Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In: Proceedings of the first workshop on evaluation and comparison of NLP systems. pp. 79–91 (2020)
[35] Ye, J., Chow, J.H., Chen, J., Zheng, Z.: Stochastic gradient boosted distributed decision trees. In: Proceedings of the 18th ACM conference on Information and knowledge management. pp. 2061–2064 (2009)
[36] Zimmet, P.Z., Magliano, D.J., Herman, W.H., Shaw, J.E.: Diabetes: a 21st century challenge. The lancet Diabetes & endocrinology **2**(1), 56–64 (2014)