Performance Analysis of Machine Learning Algorithms on Small Datasets that Includes Features from K-Nearest Neighbor Graph

Elena Ilievska and Petar Sekuloski
Faculty of Computer Science and Engineering
Ss. Cyril and Methodius University
Skopje, North Macedonia

ilievska.elena.1@students.finki.ukim.mk
petar.sekuloski@finki.ukim.mk

***Abstract*—Modern technology in today's world is largely driven by machine learning algorithms. They are incorporated into every field. Big data is not always available to us, though. We frequently have to work with limited-size of data. The purpose of this paper is to demonstrate several machine learning algorithms and their accuracy on small numerical datasets. We investigate the effectiveness of these algorithms with and without the implementation of two variables, degree and closeness centrality, which are extracted from the dataset using the k-nearest neighbor graph.**

***Index Terms*—machine learning algorithms, numeric datasets, k-nearest neighbor graph**

I. Introduction

With today's cutting-edge technology, we can predict future events in a wide range of situations. Data science and predictive analytics have become indispensable tools for improving performances across several sectors.

The fusion of healthcare and advanced technology like artificial intelligence (AI) has tremendous potential that brings more accurate and faster diagnoses, minimized medical mistakes and improved decision-making.

In this research, we examine the performance of three learning algorithms and investigate whether we can enhance their performance.

Two methods are used to get our results.

- **Approach A:** The first approach consists of preprocessing the data and modifying the algorithmic parameters to achieve optimal performance.
- **Approach B:** The second method follows the same procedures as the first method, but we've additionally included a function that calculates the k-nearest neighbor graph based on connectivity. By doing this, we obtain two additional features on top of our original dataset: *closeness* and *degree centrality*. The degree centrality feature shows how significant each data point is in the network, while the closeness feature shows how close each data point is to its closest neighbors. This additional information can hold the solution to a more effective performance.

To obtain accurate findings, we are incorporating two small datasets in our study. The first dataset's objective is to determine a patient's possibility of having diabetes. Indian women from a specific tribe who are at least 21 years old are the dataset's target audience. The objective of the second dataset is to identify early heart disease. The datasets acquired for this research come from the following sources:

- Kaggle **1**
- IEEEDataPort **2**

II. Related work

Obtaining vast amounts of high-quality data can be difficult, especially in domains such as healthcare, where access to medical information may be limited. Because of this, researchers often have to work with limited datasets that may not fully capture the big picture. Despite these challenges, machine learning (ML) algorithms must be constantly improved in order to maintain their accuracy and performance in a rapidly growing environment. Researchers have created a number of methods for utilizing datasets to overcome these difficulties, such as applying persistent homology, data augmentation (DA), feature selection (FS), etc. The use of persistent homology is one method for enhancing such ML methods. Topological data analysis (TDA) offers insightful understanding of the fundamental form and structure of complex, high-dimensional datasets **3**, **4**. Data augmentation (DA) is an alternate method to boost performance on a limited dataset **5**. By adding new samples to the dataset by modifications like scaling, rotation, or noise addition, the dataset may be made larger. This increases the diversity of the data and can help the algorithm's performance on unseen data. On the other hand, feature selection (FS) is another option. It is a method that selects and keeps the information that is most important, and removes the rest. The benefit of using FS is that it lowers the risk of overfitting and boosts the model's interpretability. **6**. There are several feature extraction methods, each having advantages and disadvantages. Among the most widely utilized methods for feature extraction are Principal Component

Analysis (PCA) and Independent Component Analysis (ICA). PCA is a technique that is commonly used for reducing the dimensionality of large datasets. It achieves this reduction by creating new variables called principal components, which are linear combinations of the original features.**7** On the other hand, ICA aims to identify statistically independent components that exhibit non-Gaussian properties and possess minimal mutual information. This makes ICA particularly useful for separating mixed signals or identifying hidden sources in complex datasets. **8**

III. Methodology

A. Dataset specifications

It is important to mention that our datasets contain only numeric and binary values. The diabetes dataset **1** has seven relevant attributes that help us determine the prediction. Below is a list of them:

- Age
- Glucose
- Blood pressure
- Insulin
- BMI - Body Mass Index
- DiabetesPedigreeFunction - determines the risk of type 2 diabetes based on family history
- Pregnancies

The heart disease dataset **2** has as well seven attributes. They are listed below:

- Age
- Chest pain type
- Resting blood pressure
- Serum cholesterol
- Maximum heart rate achieved
- Oldpeak
- ST slope

B. Data preparation

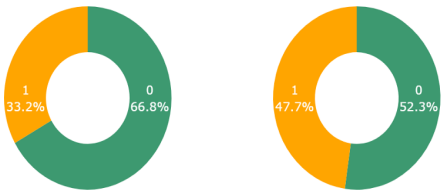
Preprocessing the data is necessary prior to working with the learning algorithms. The data must be preprocessed for a number of reasons: data scaling, data quality, etc. Duplicate rows must be removed in order to prevent the analysis from producing biased findings due to an overrepresentation of certain cases. Any single-valued columns and rows with values of NaN (not a number) or without the target feature must be removed. Each dataset only contains one file in .csv format, therefore the training and testing sets of data must be divided accordingly. Both datasets have just one target feature that stores the predictions. The values are binary and read as follows:

- 0 - no indication of the disease
- 1 - indication of the disease

Fig.1 and Fig.2 show that both datasets are categorized as imbalanced. This indicates that one class has a much higher number of instances than the other. Therefore, we must ensure that the minority class receives more attention during the training process by setting the parameter responsible for the class weight accordingly.

C. Data visualization

Fig.1 shows information about the distribution of the target value in the datasets. In terms of comparison, the first dataset is more imbalanced than the second.



(a) Diabetes dataset (b) Heart disease dataset

Fig. 1: Target feature distribution in both datasets

D. Machine learning algorithms

For our research, we used 3 different ML algorithms:

- RandomForest [RFC] - a method of ensemble learning based on decision trees. It creates a more precise and reliable model for classification problems by combining several decision trees. This method benefits small numeric datasets due to its natural ability to control noise and prevent overfitting.
- XGBoost [XGB] - optimized gradient boosting algorithm. It is popular due to its greater accuracy, scalability, speed, handling of missing data, and support for parallel computing. Therefore, it is appropriate for a range of dataset sizes, including small numeric datasets.
- LGBMBoost [LGBM] - gradient boosting-based algorithm that uses a unique tree-building technique to improve performance.

We had to execute parameter adjustments on the ML algorithms in order to achieve optimal performance.

IV. Results

Both datasets were split into two parts, one for training and one for testing. Training data was made of 70% of the original dataset, while testing data was 30%.

It is crucial to mention that we experimented with several different data splitting schemes between the training and testing sets but ultimately decided to go with the above split ratio. In the tables below, we will show the accuracy of both training and testing data using the above mentioned algorithms and the two approaches. The precision, recall, and F1-score will be shown only on the testing data.

We will focus on the relative improvement in percent on the testing dataset. Our intention was to make the testing phase better. Using features from k-nearest neighbor graph increased test accuracy for all three ML models.

We are measuring the performance by comparing the relative improvement between approaches A and B as seen in the formula below:

$$\text{Rel. imp. } (\%) = \frac{\text{Approach B} - \text{Approach A}}{\text{Approach A}} \times 100\% \quad (1)$$

- Results from the diabetes dataset:

TABLE I: Accuracy on both training and testing dataset

	RFC		**XGB**		**LGBM**	
Approach	Train	**Test**	Train	**Test**	Train	**Test**
A	0.9197	**0.8136**	0.9854	**0.7966**	0.9927	**0.7372**
B	0.9124	**0.8390**	0.9781	**0.8305**	1	**0.7796**
R. imp. (%)		**3.12%**		**4.26%**		**5.75%**

TABLE II: Precision, recall, F1-score on testing dataset

RFC			
Approach	Precision	Recall	F1-score
A	0.6750	0.7500	0.7105
B	0.6977	0.8333	0.7595
Improvement	0.0227	0.0833	0.049
R. imp. (%)	3.36%	11.11%	6.90%
XGB			
Approach	Precision	Recall	F1-score
A	0.6429	0.7500	0.6923
B	0.7143	0.7895	0.7500
Improvement	0.0714	0.0395	0.0577
R. imp. (%)	11.11%	5.27%	8.33%
LGBM			
Approach	Precision	Recall	F1-score
A	0.6429	0.6279	0.6353
B	0.7179	0.6512	0.6829
Improvement	0.0750	0.0233	0.0476
R. imp. (%)	11.67%	3.71%	7.49%

The most noticeable boost was shown in the LGBM model, which had a 5.75% improvement. Followed by the XGB at 4.26% and the RFC at 3.12%.

When comparing Approach B to Approach A, all three models show improvements in precision, recall, and F1-score. The RFC had the largest improvement in recall (11.11%), while the LGBM model had the highest improve in precision (11.67%). The XGB had the greatest improvement in F1-score (8.33%).

- Results from the heart disease dataset:

TABLE III: Accuracy on both training and testing dataset

	RFC		**XGB**		**LGBM**	
Approach	Train	**Test**	Train	**Test**	Train	**Test**
A	0.8755	**0.8080**	0.9080	**0.8036**	0.9984	**0.8405**
B	0.8831	**0.8348**	0.9176	**0.8214**	1	**0.8586**
R. imp. (%)		**3.32%**		**2.22%**		**2.15%**

The testing dataset's best improvement in accuracy was recorded by the RFC (3.32%), followed by the the XGB (2.22%) and the LGBM (2.15%). The LGBM had the largest improvement in recall (6.92%), while the RFC model had the highest improvement in precision (4.93%) and F1-score (3.15%).

TABLE IV: Precision, recall, F1-score on testing dataset

RFC			
Approach	Precision	Recall	F1-score
A	0.7500	0.8485	0.7962
B	0.7870	0.8586	0.8213
Improvement	0.0370	0.0101	0.0251
R. imp. (%)	4.93%	1.19%	3.15%
XGB			
Approach	Precision	Recall	F1-score
A	0.7477	0.8384	0.7905
B	0.7757	0.8384	0.8058
Improvement	0.028	0	0.0153
R. imp. (%)	3.74%	0%	1.94%
LGBM			
Approach	Precision	Recall	F1-score
A	0.8784	0.8333	0.8553
B	0.8634	0.8910	0.8770
Improvement	-0.0150	0.0577	0.0217
Rel. imp. (%)	-1.71%	6.92%	2.54%

V. Conclusion

Based on our research, the inclusion of additional features extracted from the k-nearest neighbor graph, can enhance the performance of the ML models on small datasets during both training and testing. On the testing model, we encountered an improvement in accuracy within a range between 2.16% and 5.75%, for precision within -1.71% to 11.67%, for recall from 0% to 11.11%, and for F1-score from 1.94% to 8.33%. Approach B provided us with a slight improvement in accuracy and other metrics by using the two additional columns (degree and closeness centrality).

The authors want to thank to Faculty of Computer Science and Engineering at the "S's. Cyril and Methodius University" in Skopje for financial support of the paper.

references