Disease prediction based on blood samples

Paavana Sai Reddy Bandi
pxb220034

Deepika Ankam
dxa210060

Haripriya Mutyala
hxm220050

Mokshagna Balabhadrapatruni
mab210007

***Abstract*—This report explores the application of machine learning methods for multi-class disease classification using blood sample data. The aim is to predict diseases using different blood markers. The study employs supervised machine learning techniques for classification and evaluates their performance using metrics like accuracy, precision, recall, and F1-score. The results are presented based on the performance of each algorithm. Future efforts will focus on improving accuracy and exploring alternative machine learning techniques.**

I. Introduction

In modern healthcare, the utilization of machine learning techniques has improved in disease diagnosis and prediction, particularly in the context of blood sample analysis. This report dives into the multi-class disease prediction based on a comprehensive dataset which has various blood markers extracted from blood samples.

The dataset was taken from the Kaggle which comprises the data of over 2000 samples. These markers include cholesterol levels, hemoglobin concentration, platelet counts, white and red blood cell parameters, hematocrit, mean corpuscular volume, mean corpuscular hemoglobin, mean corpuscular hemoglobin concentration, insulin levels, BMI, blood pressure readings, triglyceride levels, HbA1c levels, LDL and HDL cholesterol levels, ALT and AST enzyme levels, heart rate, creatinine levels, troponin concentrations, and C-reactive protein levels. The prediction column which is Disease that has 6 classes that are ”Diabetes”, ”Anemia”, ”Healthy”, ”Thalasse”, ”Thromboc”, ”Heart Disease”.

The dataset is normalized and do not have any missing values which is a good start for the modelling. All the classes are near balanced in the dataset. The diseases or label encoded for further usage in the model prediction. As part of analysis we have observed from the box plot that the insulin is more inclined for the Diabetes disease. And also the top 5 correlated features to the disease are ”White blood cells”, ”Red blood cells”, ”Hematocrit”, ”Hemoglobin”, ”ALT”.

The data is split in the ratio of 8:2 and used for training and testing respectively.

II. DATA VISUALIZATION

Examining the Fig.2 among features reveals a substantial correlation coefficient of 0.36 between ”Glucose” and ”BMI”.

we have considered few other features such as LDL Cholesterol, HDL Cholesterol, BMI, Insulin to find the correlation with the diseases.

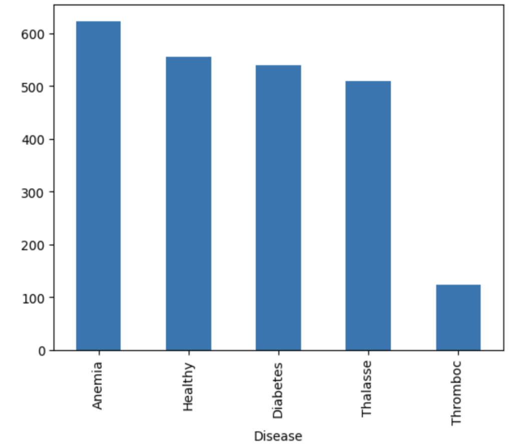


Fig. 1. Target variable

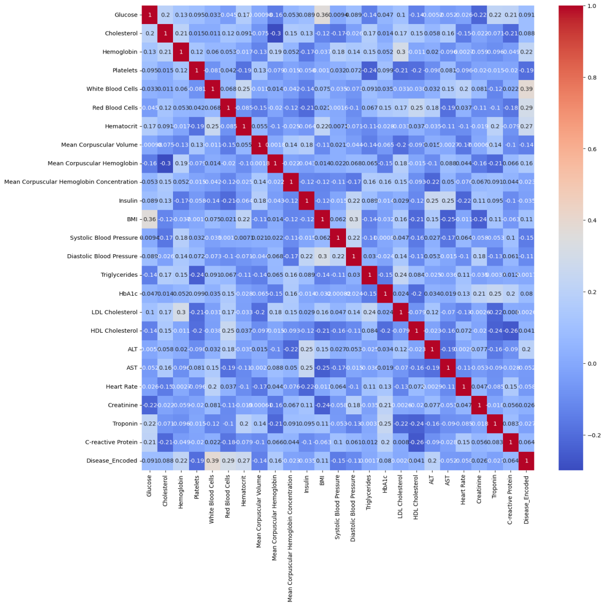


Fig. 2. Heat map

III. Performance and Analysis of Algorithms

- *Gradient Boosting Machines*
- *Support Vector Machines*

A. Gradient Boosting Machines

- Gradient boosting machines represent a group of advanced machine-learning methods that have demonstrated significant effectiveness across various real-world applications. They offer a high degree of customization to meet specific requirements of different tasks, including the ability to be tailored through various loss functions.

- Gradient Boosting Machines (GBM) revolves around the combination of boosting and decision trees to create a strong predictive model from multiple weak learners.
- The algorithm starts with an initial model, usually a naive prediction such as the log odds of each class (if using log loss). This could be a constant value like the proportion of each class in the training set.
- For each iteration, the algorithm builds a decision tree (or any weak learner) to fit the negative gradient of the loss function (pseudo-residuals). In the multiclass scenario, this involves building a separate tree for each class to predict the residuals of that class.
- After a tree is built for each class, the model updates its predictions based on these trees. The predictions from the new trees are combined with the previous predictions, scaled by a learning rate (shrinkage factor). This update is done using the formula:

$$F_k(x) = F_{k-1}(x) + \text{learning rate} \times h_k(x) \tag{1}$$

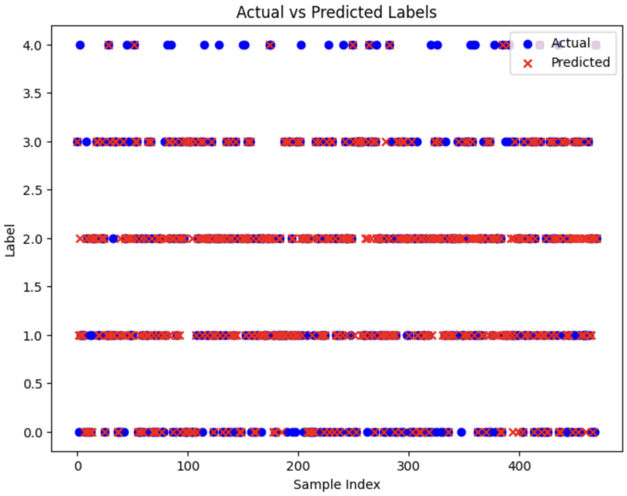
- This method of handling multiclass classification (in out case predicting diseases) ensures that GBM can effectively deal with multiple classes by focusing iteratively on the hardest class to model in the context of all others, thereby enhancing the overall predictive accuracy of the model.

How GBM is useful for our dataset (Disease Prediction):

- The features in our dataset likely have intricate relationships with the target variable (Disease). GBM, with its decision trees, is excellent at capturing such complex relationships and interactions between variables.
- With many features, GBM can assess the importance of each one in relation to disease prediction, helping to identify the most significant predictors. This is crucial for understanding which measurements are most indicative of specific diseases.
- Although our dataset is numerically consistent due to normalization, GBM's flexibility with different types of data means it would also perform well if additional categorical or non-normalized data were included.
- GBM is known for its high accuracy, especially useful in medical diagnostics where the cost of a wrong prediction can be high. The iterative correction of errors enables GBM to improve its predictions significantly with each round, focusing more on harder-to-classify instances.
- Depending on the diseases being predicted and their prevalence, there might be a possibility of imbalanced classes where some diseases are much rarer than others. GBM allows customization of the loss function to give more weight to less frequent classes, improving model sensitivity in these cases.
- GBM will provide a ranking of feature importance, which can inform clinical understanding of the diseases and potentially lead to more focused diagnostic tests or treatments based on the most influential indicators.

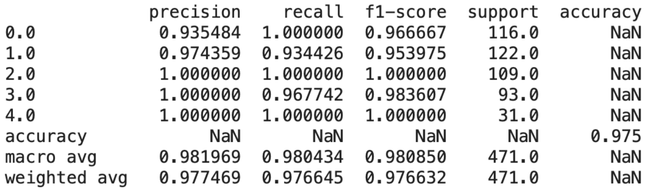
Graph between Actual and Predicted labels when

- **Learning rate:0.1**
- **Dept=1**
- **Estimators=10**



The confusion matrix allows for a detailed assessment of the model's performance, and from these values, various metrics can be calculated, including:

- Accuracy: The overall correctness of the model, calculated as (TN + TP) / (TN + TP + FN + FP).
- Precision: The accuracy of the positive predictions, calculated as TP / (FP + TP). Precision provides insights into the model's ability to avoid false positive predictions.
- Recall (Sensitivity or True Positive Rate): The proportion of actual positives correctly predicted by the model, calculated as TP / (FP + FN). Recall indicates the model's ability to capture all positive instances.
- F1 Score: The harmonic mean of precision and recall, providing a balanced measure between the two metrics. It is calculated as 2 * (Precision * Recall) / (Precision + Recall).



```
              precision    recall  f1-score  support  accuracy
0.0            0.935484  1.000000  0.966667    116.0       NaN
1.0            0.974359  0.934426  0.953975    122.0       NaN
2.0            1.000000  1.000000  1.000000    109.0       NaN
3.0            1.000000  0.967742  0.983607     93.0       NaN
4.0            1.000000  1.000000  1.000000     31.0       NaN
accuracy            NaN       NaN       NaN      NaN     0.975
macro avg      0.981969  0.980434  0.980850    471.0       NaN
weighted avg   0.977469  0.976645  0.976632    471.0       NaN
```

Fig. 3. Evaluation Matrix for Gradient Boosting Machines

B. SVM- Support Vector Machines

- Support Vector Machine (SVM): A supervised machine learning technique suitable for classification and regression tasks.
- Hyperplane Separation: SVM aims to discover a hyperplane that separates data points into distinct classes. In binary classification, the hyperplane should maximize the distance (margin) between itself and the nearest data points from each class.
- Support Vectors: The data points nearest to the decision boundary are known as support vectors. They are crucial in defining the optimal hyperplane.
- Margin: The distance between the hyperplane and the nearest data points from each class. SVM aims to maximize this margin for robust classification.
- Complex Decision Boundaries: With non-linear kernels, SVM can form intricate decision boundaries, enabling the model to identify complex data patterns.

Multi-Class SVM: Since SVMs are inherently binary classifiers, different strategies extend them to multi-class problems. The two most common approaches are:

One-vs-Rest (OvR): Each class is trained against the rest of the dataset, leading to multiple binary classifiers. The classifier with the highest confidence is chosen.
One-vs-One (OvO): A binary classifier is trained for every pair of classes, resulting in many classifiers. The final class is determined by a voting scheme.

In our implementation of a multi-class Support Vector Machine (SVM), the One-vs-Rest (OvR) strategy was employed to extend the inherently binary classification mechanism to handle multiple classes. In this approach, a separate classifier is trained for each class against the remaining dataset, creating multiple binary classifiers. The class with the highest decision function value is then chosen as the final prediction, providing the class with the highest confidence. Our core LinearSVM class was designed to minimize the hinge loss, a differentiable approximation of zero-one loss, using gradient descent to optimize the weights and bias. The classification condition is margin-based, meaning it adjusts weights and bias based on whether a data point meets or exceeds the desired margin.

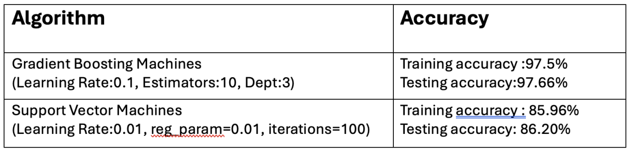
Our MultiClassSVM class aggregates the results of the various binary classifiers, predicting the class with the highest score among them. The use of Resilient Distributed Datasets (RDDs) ensures the scalability and efficiency of this distributed algorithm, enabling it to process large-scale datasets seamlessly. By leveraging the principles of hyperplane separation and margin maximization, our SVM model offers a solid foundation for handling classification tasks. Despite the complexities of multi-class SVMs, our implementation remains efficient.

	Training	Testing
Accuracy	85.96%	86.2%
Precision	86%	86%
Recall	85.95%	86.19%
F1-score	85.95%	86.19%

The SVM model has demonstrated promising results, achieving an overall training accuracy of 85.96% and testing accuracy of 86.20%. These high accuracy levels indicate the model's effectiveness in generalizing well from the training data to unseen test data. The precision, recall, and F1-score metrics for both training and testing phases are notably consistent, each being approximately 0.86, which further validates the model's robustness in classification tasks. These metrics are particularly important as they signify not only the model's ability to correctly identify positive instances (precision) but also its capacity to detect most positive cases available (recall), thereby providing a balanced measure of the model's performance through the F1-score.

IV. Results and Analysis

Below are the results of the 2 algorithms



Algorithm	Accuracy
Gradient Boosting Machines (Learning Rate:0.1, Estimators:10, Dept:3)	Training accuracy :97.5% Testing accuracy:97.66%
Support Vector Machines (Learning Rate:0.01, reg_param=0.01, iterations=100)	Training accuracy : 85.96% Testing accuracy: 86.20%

Fig. 4. Algorithm and their accuracy's

- *For GBM*, both training and testing accuracies are very high (above 97%),which suggests that the model is performing exceptionally well on this dataset. The GBM is effectively capturing the underlying patterns in the data.
- The model generalizes very well from the training set to the testing set, as evidenced by the testing accuracy being slightly higher than the training accuracy. This indicates that the model is not overfitting, despite the high complexity typically associated with GBM.
- The chosen parameters, including a moderate learning rate and a small number of estimators, seem to be very effective for this particular dataset. The depth of 3 allows the model to capture complex patterns without becoming overly specific to the training data.
- *For SVM*,both training and testing accuracies are moderate. While not as high as GBM, an accuracy around 86% is still respectable and could be suitable for many applications, depending on the criticality of errors.
- Similar to GBM, SVM shows good generalization capabilities. The testing accuracy is slightly higher than the training accuracy, which often indicates that the model is neither overfitting nor underfitting significantly.
- The low learning rate and regularization parameter, along with a sufficient number of iterations, provide a balance

between learning sufficiently complex patterns and maintaining generalizability. However, there may be room to improve performance by tuning these parameters further, possibly increasing the regularization parameter or adjusting the kernel if it's being used.

- While SVM does not reach the same level of accuracy as GBM, it still performs adequately and offers a simpler, possibly more interpretable model depending on the kernel used. It could be a good choice for scenarios where interpretability and model simplicity are more important.

V. Conclusion

- In this study, we employed two different machine learning models, Gradient Boosting Machines (GBM) and Support Vector Machines (SVM), to predict multiple diseases using a comprehensive dataset of blood samples. The GBM model, configured with a learning rate of 0.1, 10 estimators, and a tree depth of 3, demonstrated superior performance, achieving an impressive testing accuracy of 97.66% and a training accuracy of 97.5%.
- This indicates that GBM is highly effective in capturing the underlying patterns in the dataset, likely due to its robustness in handling various feature types and interactions efficiently.
- Conversely, the SVM model, despite being set with a conservative learning rate of 0.01, a regularization parameter of 0.01, and 100 iterations, achieved moderate accuracy levels of 85.96% in training and 86.20% in testing. This performance, while respectable, suggests that SVM may require further tuning and possibly a different kernel approach to improve its ability to handle the complexity of the dataset.
- The differences in performance between these models highlight the importance of model selection and tuning in predictive analytics, particularly in the high-stakes field of medical diagnostics, where the cost of misprediction can be substantial.

VI. Future Work

- Further tuning the parameters of both models could potentially enhance their predictive accuracy. For SVM, exploring different kernels (such as polynomial or radial basis function) might yield improvements, especially in capturing nonlinear relationships.
- Investigating the creation of new features or transforming existing features could provide new insights and improve model accuracy. Feature selection techniques could also be employed to reduce dimensionality and focus on the most informative attributes.
- Employing rigorous cross-validation techniques to ensure the models are robust and generalize well across different subsets of the data. This would help in affirming the models' reliability before practical deployment.
- Collaborating with medical experts to validate the clinical relevance of the predictions and adjust the models according to clinical feedback could ensure the models are practical and effective in a real-world setting.

References

[1] https://spark.apache.org/docs/latest/api /python/reference/api/pyspark.mllib.classification.SVMModel.html

[2] https://spark.apache.org/docs/2.2.0/ml-classification-regression.htmlgradient-boosted-tree-classifier

[3] https://randomrealizations.com/posts/gradient-boosting-machine-from-scratch/

[4] https://blog.paperspace.com/gradient-boosting-for-classification/

[5] https://monkeylearn.com/blog/introduction-to-support-vector-machines-svm/

[6] https://www.gormanalysis.com/blog/gradient-boosting-explained/