FINAL PROJECT REPORT

MACHINE LEARNING (MII212402)

THALASSEMIA AND ANEMIA BLOOD TEST CLASSIFICATION

Project Members:

- **Reynard Pradhitya** **(21/472680/PA/20321)**
- **Muhammad Alfi Ramadhan** **(21/472839/PA/20345)**
- **Reza Aurelio Brilliansah** **(21/475039/PA/20515)**
- **Muhammad Zaky Firdaus** **(21/477171/PA/20637)**

UNIVERSITAS GADJAH MADA

1. INTRODUCTION

1.1. Background

Anemia and thalassemia are hematological disorders characterized by a reduced ability of the blood to carry oxygen due to decreased levels of red blood cells or abnormal hemoglobin production (NCBI, 2022). The accurate differentiation between different types of anemia and thalassemia is essential for providing appropriate treatment and minimizing potential complications. Traditional diagnostic methods, such as blood tests and microscopic examination, are time-consuming, subjective, and prone to human error. In contrast, machine learning algorithms offer the potential to automate and enhance the diagnostic process, providing clinicians with efficient and reliable decision support tools.

This project is done as the final project of IUP Machine Learning class. Our task is to create a machine learning model from the given dataset. This dataset is a real life dataset taken from a hospital. Our goal of this project is to utilize this given dataset and successfully create a machine learning model with high accuracy using the methods that we have learned throughout the semester.

1.2. Dataset

Dataset used in this project is patients' blood test data from Dr. Sardjito General Hospital, Yogyakarta which contains 129 pre-splitted training data and 65 testing data. Original dataset contains three classes but for the sake of this project, only two classes are classified. No. 1 label in the dataset represents Thalassemia class, while no. 2 represents Anemia class. Training dataset has a fairly balanced number of classes, with a proportion of Anemia and Thalassemia classes of 1 : 1.04. Therefore, oversampling or undersampling is not necessary for this dataset.

Class	Training Data	Testing Data
1	43	22
2	70	34

Table 1. Dataset Variable Count

Dataset also has 19 features/attributes and 1 label column. The 19 attributes contain information about the blood test of the patient.*Table 2. Dataset Features and Attributes*

No	Features	Abbreviation	Explanation	Data Type
1	RBC	Red blood cell		float64
2	HGB	Hemoglobin		float64
3	Cellular	Cellular		float64
4	HCT	Hematocrit test	Percentage of red blood cells	float64
5	MCV	Mean corpuscular volume	Average volume of red blood cells	float64
6	MCH	Mean corpuscular hemoglobin	Average amount of hemoglobin per red blood cell	float64
7	MCHC	Mean corpuscular hemoglobin concentration	Average concentration of hemoglobin in a sample of red blood cells	float64
8	CHCM	Cellular hemoglobin concentration mean	Mean of optically measured hemoglobin within a cell	float64
9	CH	Cholesterol		float64
10	RDW	Red cell distribution width	Variability of the size of the red blood cells	float64
11	HDW	Hemoglobin distribution width	Heterogeneity of hemoglobin concentration in RBC	float64
12	Retic %	Reticulocyte	Percentage of reticulocyte in sample	float64
13	CHr	Reticulocyte hemoglobin content		float64
14	CHm	Corpuscular Hemoglobin		float64
15	RBC %	Red blood cell		float64
16	RBC %	Red blood cell		float64
17	RBC %	Red blood cell		float64
18	RBC %	Red blood cell		float64
19	RBC	Red blood cell		float64

2. METHODS

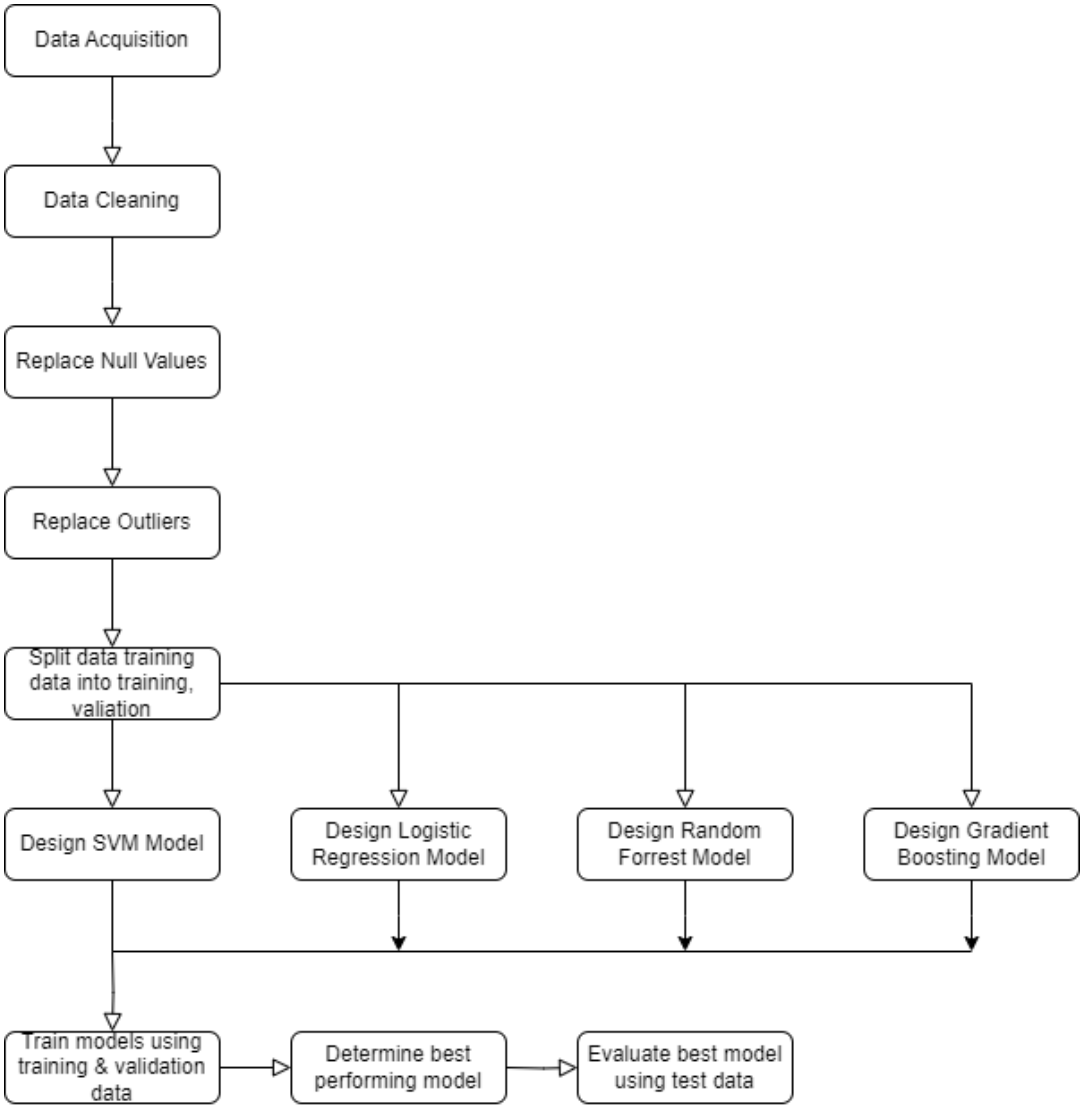

Figure 1. Flowchart of the Methodology

2.1. Data Preprocessing

2.1.1. Data Cleaning

Data cleaning involves steps to convert and format the data from the .xlsx file into a usable dataframe. Firstly, values using a comma decimal separator were converted from to utilize period as the decimal separator. Next, rows with a diagnosis value of 3 were dropped. Finally Column headers were reorganized and formatted, and the 'No.' column which served as an identifier column, was dropped.

2.1.2. Handling Missing Values

Since the dataset contained many null or missing values, the next step is to handle these values. This is especially important as the dataset size is very limited and simply removing rows with missing values might negatively affect the model training. To handle these values, the average of each column was first computed and stored into a variable. Each missing value will be replaced with the average of the column previously calculated.

2.1.3. Handling Outliers

Due to the limited size of the dataset, having outliers might greatly affect the performance of the models trained. As such, it is necessary to detect and remove outliers. First, the standard deviation for each column is calculated. A threshold for every column is then calculated, that being the twice of the standard deviation. Values that exceed this threshold will be replaced with the average of the column.

2.1.4. Dataset Splitting

We split the original training data into two more sets, those being the training set and the validation set, which are going to be used for training the model. A 80:20 split was utilized to split the training data. This leaves us with 3 sets of data in total, training set, validation set, and testing set.

2.2. Designing Machine Learning Models

```
models = {
    'SVM': (SVC(), {'C': [0.1, 1, 10, 100u], 'kernel': ['linear', 'rbf']}),
    'Logistic Regression': (LogisticRegression(), {'C': [0.1, 1, 10, 100], 'penalty': ['l1', 'l2']}),
    'Random Forest': (RandomForestClassifier(), {'n_estimators': [100, 200, 300, 400, 500], 'max_depth': [None, 5, 10]}),
    'Gradient Boosting': (GradientBoostingClassifier(), {'n_estimators': [100, 200, 300, 400, 500], 'learning_rate': [0.1, 0.01, 0.001]})
}
```

Figure 2. Summary of the Models and their hyperparameters

In order to determine the most appropriate model, several algorithms were tested, those being; SVM, Logistic Regression, Random Forest, and Gradient Boosting. To compare these models, GridsearchCV was utilized. Using Gridsearch, not only can different algorithms be tested at once, but hyperparameter tuning can also be performed to achieve the best results. Figure 2 summarizes the models as well as the parameters that were used in Gridsearch. Through this process, the best and most accurate model, as well as their parameters, can be found.

2.3. Final Model Evaluation

Once the final model is obtained, the next step is to test its performance using the testing set. Firstly, the data needs to be preprocessed by performing data cleaning and null value handling. The final model is then given the testing data and makes predictions for the diagnosis, These predictions can then be compared with the actual diagnosis labels in the testing set. This is done by creating a classification report, which will show the accuracy, precision, recall, and f1-score of the model. The formula of the evaluation metrics are presented below:

$$accuracy\ (acc) = \frac{TP + TN}{TP + TN + FP + FN} \qquad recall\ (r) = \frac{TP}{TP + FN}$$

$$precision\ (p) = \frac{TP}{TP + FP} \qquad F1 - measure\ (Fm) = \frac{2pr}{p + r}$$

3. EXPERIMENTAL RESULTS & ANALYSIS

3.1. Exploratory Data Analysis

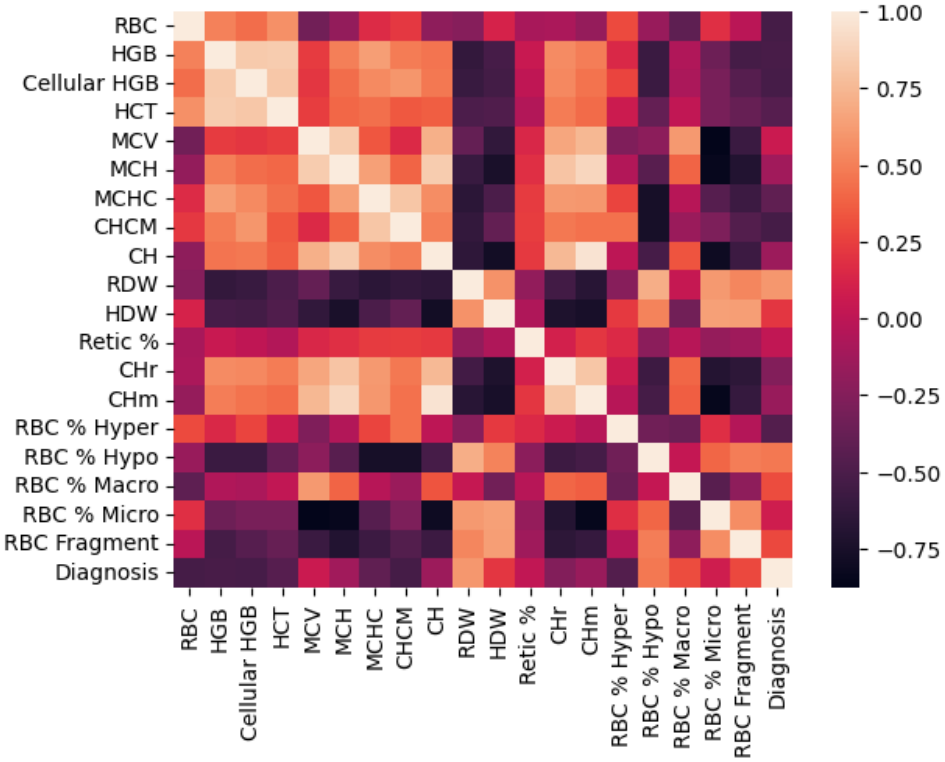

Figure 3. Heatmap of the Variables

3.2. Results

3.2.1. Model Comparison

Model	Training Score	Validation Score
SVM	**0.86**	**0.95**
Logistic Regression	**0.85**	**0.86**
Random Forest	**0.87**	**0.82**
Gradient Boosting	**0.86**	**0.73**

Table 3. Gridsearch Results Summary

Using GridSearchCV, the best hyperparameters from each model is obtained. SVM (Support Vector Machine) has the C hyperparameter of 100 and kernel of Radial Basis Function (RBF) which results in training accuracy of 86% and validation of 95%. Logistic regression has C hyperparameter of 1 and penalty 12, resulting in training score of 85% and validation of 86%. Random Forest has max_depth hyperparameter of 10 and n_estimators 100, resulting in a training score of 87% and validation of 82%. Gradient Boosting has the learning rate hyperparameter of 0.001 and n_estimators of 100, resulting in a training score of 86% and validation of 73%.

3.2.2. Model Testing

After comparing the training and validation accuracy of SVM, Logistic Regression, Random Forest, and Gradient Boosting, SVM has the highest training and validation score. Therefore, it was decided that we would use the SVM model for the testing data.

Class	Precision	Recall	F1-Score
1	0.83	0.91	0.87
2	0.94	0.88	0.91
Macro Avg	0.89	0.90	0.89
Weighted Avg	0.90	0.89	0.90
Accuracy			**0.89**

Table 4. SVM Accuracy

	1 (Predicted)	2 (Predicted)
1 (Actual)	20	2
2 (Actual)	4	30

Table 5. Confusion Matrix

The above confusion matrix reveals that the model accurately classified 20 instances as Thalassemia and 30 instances as Anemia, reflecting our SVM model's accuracy of 0.89. However, the model made 2 misclassifications as Anemia and 4 misclassifications as Thalassemia.

3.2.3. Analysis & Discussion

Among the four models tested in this paper; SVM, Logistic Regression, Random Forest, and Gradient Boosting, the best performing model was determined to be SVM. Although the training scores of the 4 models were very similar, the validation score of SVM was significantly higher compared to the other models. There are several reasons why SVM might have achieved the highest score. For instance, SVM works well in scenarios with a high number of features, which in this case is 19. In addition, SVM is able to generalize better than the other models since it is effective even with a limited number of training data.

The final model was able to successfully achieve an accuracy of 89%, which is relatively high. However for medical purposes, the model might still need to be improved and fine-tuned further in order to achieve an accuracy closer to 100%. There are several limitations to this paper that could be addressed in order to obtain a better model. For example, cross validation could be implemented to address the limited number of data available. The parameters used in gridsearch are also somewhat limited, and more parameters could be added to potentially discover better models.

4. CONCLUSION

This report focused on utilizing machine learning algorithms to develop a model for the accurate diagnosis of anemia and thalassemia based on blood test data. The traditional diagnostic methods for these hematological disorders are time-consuming and subjective, making machine learning an alternative approach for automating and enhancing the diagnostic process. After further analysis of the real life dataset, the best model to use is SVM. Based on the result of the testing above, the best method to classify the blood testing is SVM with an accuracy of 89%. Despite the high performance, there were a few misclassifications observed, which could be further improved through fine-tuning and addressing limitations.

Through the development of the SVM demonstrated the potential of machine learning algorithms, specifically SVM, in aiding the diagnosis of anemia and thalassemia based on blood test data. Further enhancements and optimizations can be explored to improve the model's accuracy and generalizability in a medical setting.

Appendix

Program and result can be found in this link:

https://colab.research.google.com/drive/1W2qiICfmIrRUUse0iTJ9c8OlM0pwKpKx?usp=sharing

References

[1] National Center for Biotechnology Information, "Endocannabinoids," in StatPearls, 2021. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK545151/. [Accessed: Jun. 17, 2023].

[2] Taylor, D. W., "Mean Corpuscular Hemoglobin Concentration (MCHC)," Verywell Health, Feb. 12, 2021. [Online]. Available: https://www.verywellhealth.com/mean-corpuscular-hemoglobin-concentration-797200. [Accessed: Jun. 17, 2023].