**Table of Contents**

[**1.** **Classification Algorithms** 2](#_Toc187006731)

[**1.2.** **Showing accuracies of each model.** 2](#_Toc187006732)

[**1.3.** **Explanation of how the two models work (include references).** 4](#_Toc187006733)

[1.3.1. Logistic Regression Model 4](#_Toc187006734)

[1.3.2. Random Forest Model 5](#_Toc187006735)

[**1.4.** **Reason of the better performance of Random Forest:** 6](#_Toc187006736)

[**1.5.** **Suggest different methods to improve your models.** 6](#_Toc187006737)

[**1.6.** **Conclusion** 7](#_Toc187006738)

[**References** 7](#_Toc187006739)

# **Classification Algorithms**

The assignment task was to run two Classification models or algorithms on a dataset and compare their results to see which one performed better and why. So, first, a dataset (Rezaei, 2023) was selected on the heart condition classification, whether the heart is healthy or not. The dataset consists of three hundred three data points with thirteen attributes defining the output column. The distribution among the outputs such as 0 and 1 is 165 and 138 respectively. From this, we can say that the data is not quite balanced but the difference is minor. The dataset (Rezaei, 2023) has the following attributes supporting the output:

1. age: age in years
2. sex: sex
3. cp: chest pain type
4. trestbps: resting blood pressure (in mm Hg on admission to the hospital)
5. chol: serum cholesterol in mg/dl
6. fbs: (fasting blood sugar > 120 mg/dl)
7. restecg: resting electrocardiographic results
8. thalach: maximum heart rate achieved
9. exang: exercise induced angina
10. oldpeak = ST depression induced by exercise relative to rest
11. slope: the slope of the peak exercise ST segment
12. ca: number of major vessels (0-3) colored by flourosopy
13. thal

After confirming the dataset, some basic data exploratory analysis was performed such as describing the data, getting missing values information of the dataset, checking whether outliers are present in the dataset or not, and how the distribution of data is. After performing the exploratory analysis some preprocessing is done such as filling in missing values, removing outliers, and some feature engineering. Now, it’s time to choose the algorithms/models to perform the classification task. Out of all the classification models, two algorithms/models were selected for the classification task, **Logistic Regression (LR)** and **Random Forest (RF)**. The data was divided into two sets for training and testing in a ratio of 80% and 20% respectively. In the below section, results are discussed briefly along with the working of the algorithms selected for the task. A section is added to examine the accuracy of both models which performs better and why it performs better. Finally, a detailed section is added to discuss the measures taken to increase the efficiency of the models.

## **Showing accuracies of each model.**

The dataset was trained and tested on both models individually and evaluated based Accuracy and F1-Score. The data was divided into an 80% to 20% ratio for training and testing, respectively.

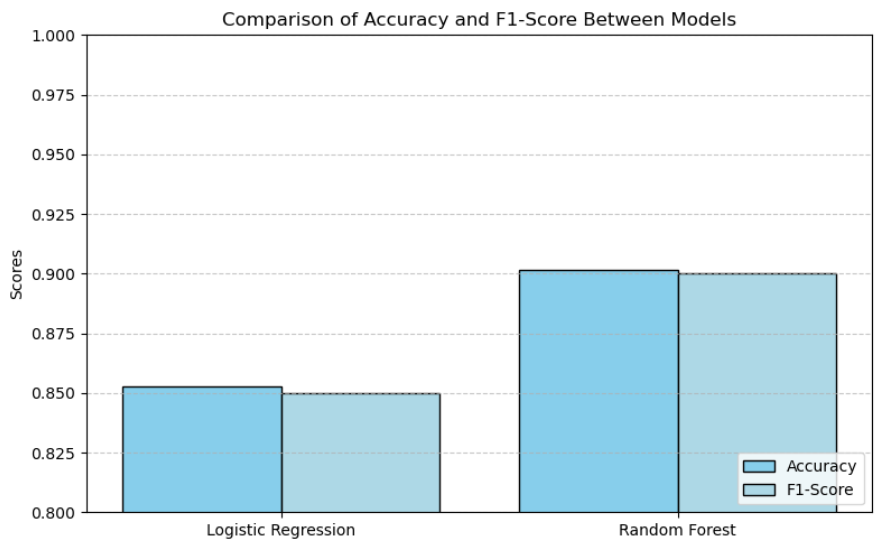
**Table 1: Logistic Regression Classification Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Class 0 | 0.81 | 0.90 | 0.85 | 29 |
| Class 1 | 0.90 | 0.81 | 0.85 | 32 |
| Accuracy |  |  | 0.85 | 61 |

**Table 2: Random Forest Classification Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | precision | recall | f1-score | support |
| Class 0 | 0.85 | 0.97 | 0.90 | 29 |
| Class 1 | 0.96 | 0.84 | 0.90 | 32 |
| Accuracy |  |  | 0.90 | 61 |

Based on the accuracies and classification reports as shown in



**Figure 1: Graph Visualizing the Results of Both Models**

***Table 1: Logistic Regression Classification* Matrix** and

***Table 2: Random Forest Classification* Matrix**, the Random Forest model performed better than Logistic Regression in this case. Random Forest achieved a higher overall accuracy (0.90 vs. 0.85) and demonstrated better precision and recall for both classes. A comparison graph showed in ***Figure 1: Graph Visualizing the Results of Both Models***, elaborated the results more clearly and showed that the Random Forest (RF) out performs the Logistic Regression (LR).

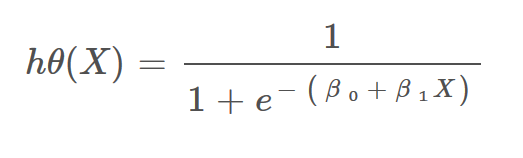
## **Explanation of how the two models work (include references).**

### Logistic Regression Model

**Main Idea:**

Logistic Regression (LR) is a statistical model used for binary classification (Anshul, 2024). It predicts the probability of a binary outcome using the logistic function also known as the sigmoid function or equation as given in ***Figure 2: Sigmoid Function or LR Equation***.

**Equation:**

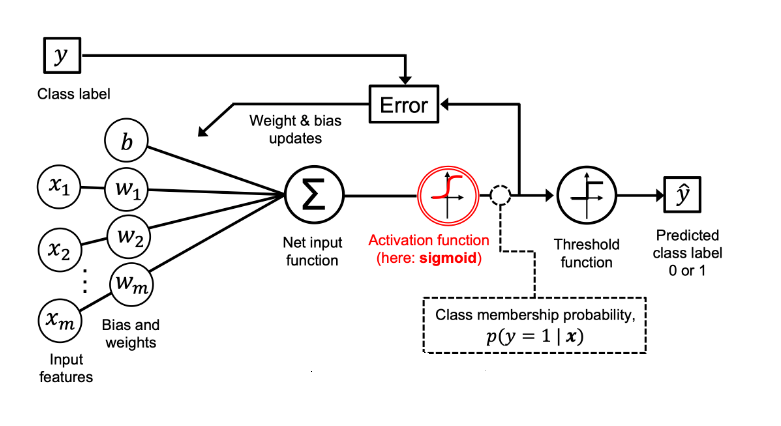


**Figure 2: Sigmoid Function or LR Equation**

This equation represents the hypothesis function for **Logistic Regression** (Anshul, 2024), often called the **Sigmoid Function**. It is used to predict probabilities for binary classification problems, where the output lies between 0 and 1. The sigmoid function outputs probabilities, which are then threshold (e.g., at 0.5) to make binary predictions. This function is the foundation of logistic regression's ability to model classification problems.

**Work Flow of the Logistic Regression:**

A diagram shown in ***Figure 3: Logistic Regression Work Flow*** depicts the work flow of **Logistic Regression (LR)** (Kumar, 2024) for binary classification. Let’s discuss the steps with minor descriptions:



**Figure 3: Logistic Regression Work Flow**

1. **Input Features (x1, x2, ..., xm)**: These are the independent variables or features of the dataset that serve as input to the model.
2. **Weights (w1, w2, ..., wm) and Bias (b)**: Each feature is assigned a weight that determines its contribution to the prediction. The bias (b) helps shift the decision boundary.
3. **Net Input Function (Σ\Sigma)**: The weighted sum of the input features and bias is calculated
4. **Activation Function (Sigmoid)**: The sigmoid function is applied to the net input to squash the output into a probability range between 0 and 1
5. **Threshold Function**: A threshold (typically 0.5) is applied to the probability to determine the predicted class label
6. **Error Calculation**: The difference between the predicted output and the actual label is calculated using a loss function, such as cross-entropy.
7. **Weight and Bias Updates**: The error is used to update the weights and bias via optimization techniques like gradient descent to minimize the loss.

This process repeats iteratively during training until the model achieves optimal performance in classifying the data. The diagram visually illustrates how logistic regression transforms input features into class probabilities and final predictions (Kumar, 2024).

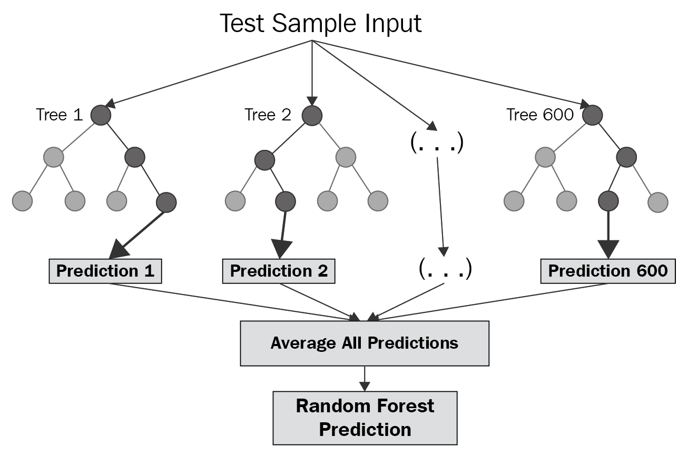
### Random Forest Model

**Main Idea:** Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

**Working of Random Forest in steps:**

A diagram illustrating the working of Random Forest is shown in ***Figure 4: Random Forest working***. There are several steps included in the working such as:

1. Data is divided and model is trained on 80 percent of the data. And on remaining 20 percent, the model is tested.



**Figure 4: Random Forest working**

1. Random forest is an extended form of decision tree, In this multiple decision trees are generated using bootstrap aggregation technique and each tree make a prediction independently.
2. The Random Forest combines the individual tree predictions: For **classification**, it performs a majority vote (or computes probabilities and chooses the class with the highest probability).
3. The aggregated result (average or majority vote) is the final prediction of the Random Forest for the test sample.
4. This ensemble approach allows Random Forest (Team, n.d.) to achieve high accuracy and robustness, especially in handling non-linear relationships and interactions among features.

## **Reason of the better performance of Random Forest:**

The performance of **Random Forest** in this dataset can be attributed to its ability to capture **non-linear relationships** between features and the target variable. Unlike Logistic Regression, Random Forest models **complex decision boundaries**, which is beneficial if the classes are not linearly separable. Additionally, Random Forest accounts for **feature interactions**, allowing it to uncover dependencies between features that Logistic Regression cannot. Furthermore, Random Forest is inherently more **robust to outliers and noise** due to its ensemble nature, where predictions are averaged across multiple trees, reducing the impact of individual errors.

In contrast, **Logistic Regression** assumes a **linear relationship** between features and the target variable, limiting its ability to handle non-linear patterns. If the dataset contains complex relationships, Logistic Regression may fail to capture them effectively. Moreover, Logistic Regression is more **sensitive to outliers**, which can skew its predictions and lower its performance compared to Random Forest. These limitations explain why Random Forest outperforms Logistic Regression on this dataset.

## **Suggest different methods to improve your models.**

To further enhance model performance, several strategies can be implemented. **Feature Engineering** plays a vital role in improving predictive power; analyzing and visualizing the dataset can help identify key features, and incorporating interaction terms or polynomial features can capture non-linear relationships. Effective **Data Preprocessing** is equally important, including normalizing or standardizing features to ensure uniformity and handling missing values appropriately to avoid introducing bias. **Model Tuning** using cross-validation is critical; for Random Forest, adjusting hyper-parameters such as the number of trees, maximum depth, and minimum sample split can significantly boost performance. Finally, **Feature Selection** helps simplify the model by removing irrelevant or redundant features. Techniques like Recursive Feature Elimination (RFE) or leveraging feature importance scores from Random Forest can be used to refine the feature set and enhance the model's efficiency and accuracy. These steps collectively aim to improve both the reliability and predictive power of the models. Two of the above-mentioned methods were used in this assignment to improve the results such as feature engineering and Model tuning.

## **Conclusion**

In this report, two classification models/algorithms such as **Logistic Regression (LR)** and **Random Forest (RF)** were evaluated and compared on the basis of performance. The results showed that the Random Forest (RF) outperforms the Logistic Regression in terms of accuracy (90% vs. 85%) and F1-score for both classes. The higher results of Random Forest (RF) can be deduced because to its capability to work on non-linear relationships. On the other hand, Logistic Regression's supposition of linear relationships between features and target variable restricted its effectiveness in detecting difficult patterns within the dataset. Moreover, its sensitivity to outliers contributed to its lower performance. The Random Forest model's flexibility in modeling non-linear decision boundaries and its resilience to data imperfections provided a significant advantage. Various improvement methods, including feature engineering, hyper parameter tuning, data scaling, and outlier detection, were employed to enhance model performance.

## **References**

Anshul, 2024. *Logistic Regression: A Comprehensive Tutorial.* s.l.:s.n.

Kumar, A., 2024. *Logistic Regression in Machine Learning: Python Example.* s.l.:s.n.

Rezaei, A., 2023. *Kaggle.* [Online]   
Available at: https://www.kaggle.com/datasets/arezaei81/heartcsv  
[Accessed 12 12 2024].

Team, C., n.d. *Random Forest.* s.l.:s.n.