Prediction of Cardiovascular Disease with Machine Learning Classification Algorithms

Lucas Oliveira
Faculty of Engineering, Environment and Computing
Coventry University
Coventry, England
Lopesoll@uni.coventry.ac.uk

Martin Tay
Faculty of Engineering, Environment and Computing
Coventry University
Coventry, England
Taym5@uni.coventry.ac.uk

***Abstract*— The objective of this research paper is to examine the prediction of heart diseases with three classification algorithms. Cardiovascular disease is one of the prevalent causes of death in the world. The healthcare industry has a significant role to play in terms of providing timely intervention to save lives of vulnerable patients and the current number of annual cases around the world in terms of statistics is still disturbing and calls for a campaign to improve the well-being of humans in countries where the rates are terribly high.**

The dataset used to perform this experiment is comprised of 253680 rows and twenty-two columns reflecting the input and target variables in the samples. The challenge brought by the imbalanced state of the dataset was resolved with the SKLearn resampling techniques in python. Each sample contains twenty-two features, which were reduce to 17 features with recursive feature elimination method. The outcome of the experiments after training the three models with the heart disease dataset yielded the respective results 78%, 81%, and 89% models for the respective classification algorithms (Decision Tree, Support Vector Machine and Logistic Regression). Other activities like hyperparameter optimization and feature selection were explored to ensure the best effort on the predicted output variables. The overall performance of the classification algorithms would be measured against the relevant evaluation metrics parameters before and after training of the models.

Keywords— Cardiovascular Disease (CVD), Decision Tree, Support Vector Machines, Logistic Regression, Confusion Matrix, Recursive Feature Elimination)

I. INTRODUCTION

Cardiovascular disease is one of the prevalent causes of death in the world and accounts for one of the most painful deaths that can be avoided under strict supervision. Cardiovascular diseases cause an average of 17.7 million deaths each year (44% of NCD fatalities) making it one of the most deserving topics for research on prevention. "Cardiovascular diseases (CVD) are a group of disorders of the heart and blood vessels which is the most significant cause of death globally[1]. Despite the critical fatality rate 90% CVD can be prevented by taking necessary precautions[2]. The impact of cardiovascular diseases can be very painful in the early years of any human being and have some ripple effects to the society due to the pain and economic effects on families and the economy of any country.

There are several risk factors that contribute to cardiovascular health conditions. It is no news that in most cases, the cause of the disease can be attributed to lifestyle and human preference of diets. Physical activities and healthy eating have been proven to help improve the health and well-being of humanity. The application of machine learning to predict cardiovascular diseases can help in the sensitization and improve the conditions of diagnosed patients. There are several case studies concerning cardiovascular diseases.

There are three types of prevention mechanisms to prevent and reduce the impacts of a disease. "Primary prevention refers to the steps taken by an individual to prevent the onset of the disease. This is achieved by maintaining a healthy lifestyle choice such as diet and exercise. "Secondary prevention focuses on reducing the impact of the disease by early diagnosis prior to any critical and permanent damage[1]. This facilitates avoiding life threatening situations and long term impairments from a disease. Tertiary prevention is used once long term effects set in, by helping the patients to manage pain, increase life expectancy, and increase the quality of life. The secondary prevention of CVD includes diagnosis and prevention. Most critical step of secondary prevention is early diagnosis which allows medical professionals to provide required care for patients and improve the quality of life. This requires identifying risk factors, criticality of risk factors, and how the variation of these factors relates to CVD . Upon early diagnosis, patients could be directed to required treatments affording a higher quality of life[1].

The inspiration for this research paper draws strength from the use of machine learning algorithms by medical professionals to make predictions of vulnerable patients and it is an improvement on the work initiated by [1]. Health improvements is tied to continuous research and implementation of state-of-the-art methodologies explored by experts to diagnose and prevent diseases like the one examined in our case study.

II. PROBLEM AND DATASET

As required, the purpose of this task is to make predictions with the independent variables and dependent variables. These independent variables contain certain information that may cause a cardiovascular condition in patients. The causes of cardiovascular disease borders around certain risk factors like blood pressure, obesity, age, sex, diet, exercise, smoking, health insurance, mental health, physical health, consumption of alcohol, rest, or sleep, and record of health check-up, etc. One major issue is the challenge with the imbalanced dataset and a resampling technique would be introduced to solve the problem. This in our case study is linked to what has been tagged as the independent features and they are outlined in the table below.

S/N	Features	Description	Data Type
1	HighBP	Patients who have been diagnosed of high blood pressure or not by a medical professional	Categorical
2	HighChol	Patients who have been informed about the presence of high cholesterol in the body or not	Categorical
3	CholCheck	Patients who have had their cholesterol level checked in the last 3 months	Categorical
4	BMI	Body Mass Index (BMI)	Numerical
5	Smoker	Patients who smoke regularly and at least a packet a day	Categorical
6	Stroke	Patients who have been diagnosed of having stroke by a medical professional	Categorical
7	AnyHealthcare	Patients who have some type of health insurance or not	Categorical
8	Diabetes	Patients who have been diagnosed of Type 1 or 2 diabetes, or never been diagnosed of any form of diabetes by a medical professional	Categorical
9	PhysActivity	Patients that carry out regular exercise or not in the last 3 months	Categorical
10	Fruits	Patients that consume fruits regularly or irregular in the last 3 months	Categorical
11	Veggies	Patients that consume veggies regularly or irregular in the last 3 months	Categorical
12	HvyAlcoholConsump	Patients that consume alcohol regularly or irregularly in the last 3 months	Categorical
13	NoDocbcCost	Patients who have had a need to see the doctor but could not due to the cost	Categorical
14	GenHlth	Patients whose health is generally good or bad	Categorical
15	MentHlth	Patients who have had mental health related issues in the last 30 days	Categorical
16	PhysHlth	Patients who have had physical health related issues in the last 30 days	Categorical
17	DiffWalk	Patients who have difficulty in walking or climbing the stairs	Categorical
18	Sex	Gender of the patient (Male = 0 or Female = 0)	Categorical
19	Age	Fourteen-level age category	Categorical
20	Education	Highest level of education completed	Categorical
21	Income	Annual household income of the patients from all sources (Yes = 1, No = 0)	Categorical
22	HeartDiseaseorAttack	Has the patient been diagnosed of cardiovascular disease or not	Categorical

Table 1 – Description of Independent and Dependent Features in the dataset

The prediction of cardiovascular disease is a classification problem being that the outcome of the prediction is a categorical response variable based on certain predictor (multiple) variables, indicating whether a patient is diagnosed with a heart condition or not diagnosed. In machine learning, there are two major learning techniques, supervised learning, and unsupervised learning. Supervised learning algorithms are trained with labelled input and output data, while unsupervised learning algorithms are trained with unlabelled variables (*see figure 1 and 2*). The problem we have at hand is that of a supervised learning algorithm and both input and output data already labelled would be examined and evaluated when loaded unto the three classification algorithms selected for this experiment.

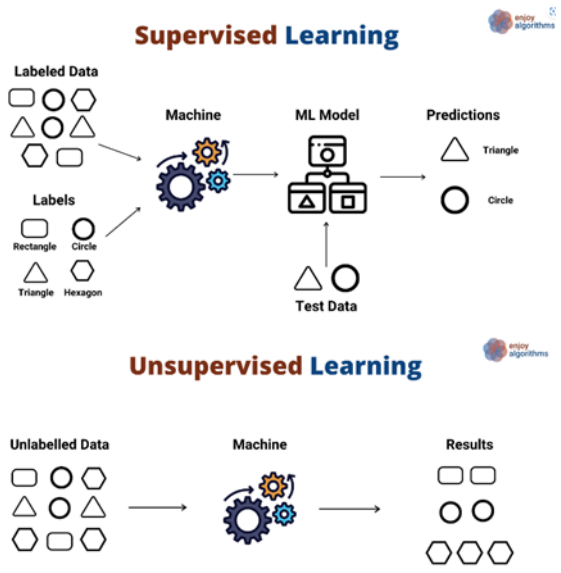

Figure 1 and 2 – Graphic display of Supervised Learning and Unsupervised Learning. NOTE: From *Supervised and Unsupervised Learning (an Intuitive Approach)* by [3]

As depicted in the diagram above, figure 1 is an illustration of the data is processed before insights are gleaned in supervised learning algorithms. Insights are drawn by from the correlation between the interactive features against the response variable. This is usually an iterative process, which calls for the model to be trained until optimal performance is achieved and the model can make accurate predictions. The process of model optimization is called hyperparameter tuning and would be discussed in detail in the next section. Figure 2 depicts the learning process of unsupervised learning and how it generates insights or patterns for analysis. Learning for unsupervised is done with unlabelled data to reveal insights and patterns in the dataset. Key difference is in how the data is learned and processed before predictions are done.

III. Methodology

A. *Data pre-processing*

Data pre-processing is a crucial step before applying machine learning that involves preparing the data for the models. This step can drastically improving performance. A technique employed in this paper was handling null values, which wasn't required because the dataset contained no missing values. Other techniques that were used and are covered in detail in this section are Resampling, to handle the label imbalance, Standardization and Feature selection.

1) Resampling

Imbalanced data makes the model predict a class more often that the other, leading to a high accuracy but poor results. This is a significant issue discussed in detail in section C. The original dataset contained imbalanced data, at a rate of 1:10 with one sample of class 1 for every ten of class 0.

Several resampling models were compared, the comparison is available in the GitHub repository. The solution that yielded the best results is a combination of Random Over Sampler and Repeated Edited Nearest Neighbours.

a) RandomOverSampler

Random over-sampling works by randomly generating instances of the minority class or under-represented class.

b) RepeatedEditedNearestNeighbours

Edited Nearest Neighbours is an algorithm that applies the nearest-neighbour and removes samples which do not agree with the neighbourhood. Two selection criteria are available all, "all" and majority, "mode". A sample is eliminated if the majority of the neighbours belong to the other class with "mode". And using "all" a sample is eliminated if any of the neighbours belong to the other class.

Repeated ENN repeats ENN multiple times resulting in the elimination of more samples. [11]

2) Standardization

Standardization is a requirement for some linear learning models such as SVM and Logistic regression. These models assume that the data is centred around 0 and have variance in the same order. If one of the features has a greater scale leads the model to attribute a higher weight to that feature limiting the model from learning from other features.

a) Standard Scaller

Standard Scaler is extremely simple, it standardizes the features by removing the mean and scaling to unit variance. The standard score of a sample (x) is calculated:

$$z = (x - u) / s$$

Where u is the mean of training samples and s is the standard deviation of the training samples [12].

3) Feature selection

Feature Selection is the process of reducing the number of features, in order to reduce the time taken to fit the algorithms and reducing the "noise" in the data, while minimizing information loss. In this paper Recursive Feature Elimination (RFE) was implemented.

a) Recursive Feature Elimination

A variation of RFE was used, RFECV applies Recursive feature elimination with cross validator to find the optimal number of features.

RFE utilizes an external estimator that assigns weights to features, such as the coefficients of a linear model, the model then recursively removes a given number of features on each iteration until the number of features to select is reached, this number is also passed to the algorithm in the parameters. [12]

B. *Machine learning models*

1) Decision Tree Classifier

Decision Tree Classification algorithm is one of the simplest techniques to implement in solving classification or regression problems. The supervised learning model learns from the class labels/interactive features by making system-based rules that guides its decision to make predictions. In the words of [4], Decision Tree learning method is a method for approximating discrete-valued target functions, in which the learned function is represented by a decision tree".

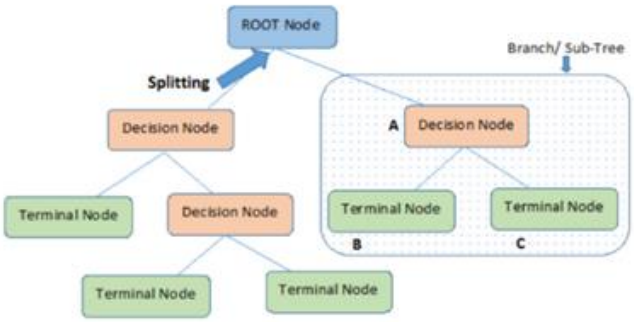

Figure 5 – A diagram of a decision learning process in Decision Tree. From Decision Tree Algorithm – A Complete Guide by [5]

The root node is where the learning process begins with several splits into finite terminal nodes before making a prediction. To make decisions before splitting, decision tree uses the amount of information gained about to improve the nodes. The information gained is measured by the entropy and mathematically expressed in the formula below, "where X is the random variable or process, Xi is the possible outcomes, and p(Xi) is the probability of the possible outcomes" [6].

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log \mathrm{P}(x_i)$$

Figure 3: NOTE: From A Complete Guide to Decision Tree Split using Information Gain, by [6].

By definition, [6], the entropy of any random variable or random process is the average level of uncertainty involved in the possible outcome of the variable or process.

2) Logistic Regression

The third classification algorithm used in this experiment is Logistic Regression. In machine learning, logistic regression uses the concept of probability to make predictions in classification problems. "The hypothesis in logistic regression tends to limit the cost function between 0 and 1" [7]. This is mathematically expressed as;

$$h\theta(X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X)}}$$

Figure 4: NOTE: The hypothesis of logistic regression. From: Introduction to Logistic Regression, by [7].

3) Suport Vector Machine

Support vector machine is a classification algorithm which performs its prediction by splitting the classes into two groups with a line mathematically positioned by two support vectors surrounding a hyperplane, typically, in a multidimensional space.

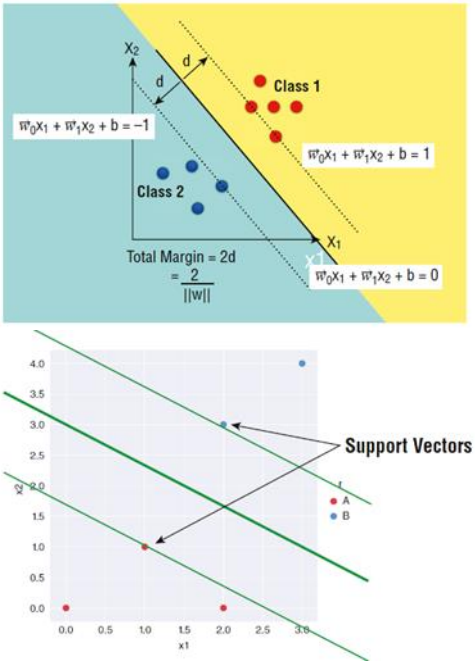

Figure 6 and 7 – Graphical display of how prediction is performed in support vector machine classification. NOTE: From Python Machine Learning, by [8]

C. *Evaluation metrics*

In a problem with imbalanced labels as this one accuracy can not be used to evaluate a model. When a model is fed extremely imbalanced data is going to predict only the majority class, resulting in near 100% accuracy even tough the model is terrible at prediction one of the classes. Therefore, other evaluation metrics must be used.

1) Confusion Matrix

Confusion matrix is a performance evaluation metric for classification problems. It shows a table representing the combinations of actual and predicted values [15].

2) Precision

Precision represents the ratio of correctly predicted true positives to the total number of values predicted positive.

Precision = TP/(TP + FP)

3) Recall

Recall is the ratio of correctly predicted true positives to the total number of positive values.

$$Recall = TP/(TP + FN)$$

4) F1-Score

F1-Score it the harmonic mean of precision and recall combining them in one single metric [16].

$$F1 = 2 \times [(Precision \times Recall) / (Precision + Recall)]$$

Is important to mention that not one single metric fits all needs, more than one must be analysed to evaluate the model.

D. Hyperparameter tunnig

Hyperparameter tunning is the process of fitting the model with a given set of parameters and return the optimal set of parameters for the given model. In this paper we used GridSearchCV to perform this task.

1) Grid search

GridSearchCV fits the given set of parameters into the given learning model and scores every combination of parameters. Then using the "best_params_" attribute the best set of parameters is returned [14].

IV. Experimental Setup

The dataset originally has 253680 rows and 22 columns. After the label imbalance issue was found the first concern was comparing resampling techniques. All the code comparing the resampling techniques is available in the git hub repository (Appendix).

A. Resampling techniques

A major issue when using resampling techniques is data leakage. Data leakage occurs when the data is resampled before being split into training and testing sets. Two issues emerge from this, the test data should be naturally imbalanced to properly test the algorithm, if the entire dataset is resampled the test set is not going to reveal potential bias of the model. The second issue is that some resampling algorithms like random over sampler or ENN utilize other samples to generate or select other samples, therefor the same information is going to be in the train and test sets (data leakage) [17].

1) Repeated ENN

The parameter number of neighbours to consider (n_neighbours) was set to 7 because the higher the number of neighbours more samples are being removed, and 7 resulted in a satisfactory ratio between both labels.

B. Feature selection

RFECV was used to find the optimal number of features. The data was then transformed using the transform method, the optimal number of features for this dataset is 17. Bellow is a plot showing the number of features against the mean test accuracy.

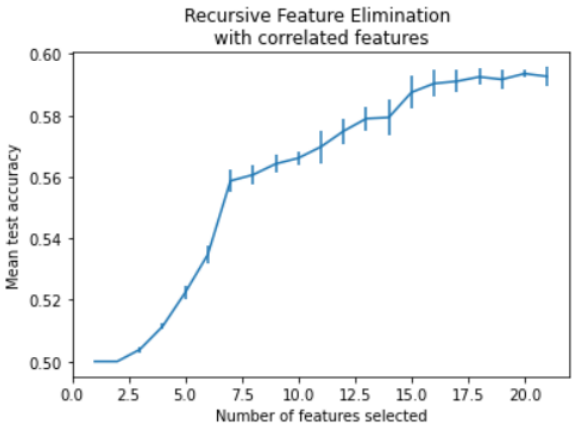

Figure 8: Mean test accuracy per Number of features plot

C. Machine learning models and hyperparameter tunning

1) Decision Tree Classifier

The parameter being tuned for this model are max_depth, criterion, and class_weight. The first indicates the maximum hight of the tree. The second is criterion, which represents the function to measure the quality of a split, the model was tested with all three available functions, Gini, entropy and log_loss, the default and most common function is Gini, however in this case entropy was selected by the grid search. The third and final parameter being tuned for this model was Class_weight which attributes weights to the classes, helping with the label imbalance issue. To get the class weights Sklearn's compute_class_weight function was used.

2) Logistic Regression

Logistic regression is a simple model there are few parameters that can be tuned. The parameters that were tuned were: Penalty, C, and Class_weight. The parameter Penalty is the type of regularization, in this case were tested two values for this parameter l2 and none, with l2 being the best fit. The parameter C is the inverse of regularization strength, smaller values specify stronger regularization [18]. The values tested for this parameter were [0.001, 0.01, 0.1, 1, 10], with a combination of l2 and C:0.001 being the preferred combination of parameters.

3) Suport Vector Machine

Support Vector Machines unlike Logistic regression is a very complex model, it has many of parameters that can be tunned to improve the model. However three parameters determine how the model behaves more than any others, these are C, Gamma, and Kernel.

Kernel is the kernel type to be used in the SVM. Kernels are used to transform the two dimensional data into higher dimensional data making it easier for the model to fit a line through the data points. The kernel selected by the GridSearch was RBF. Gamma is the coefficient for the RBF function, this value controls the hardness of the kernel. High values of gamma result in overfitting and values of gamma to low result in lines too straight leading to poor results. In the figures below there is a representation of two decision boundaries with the different levels of Gamma.

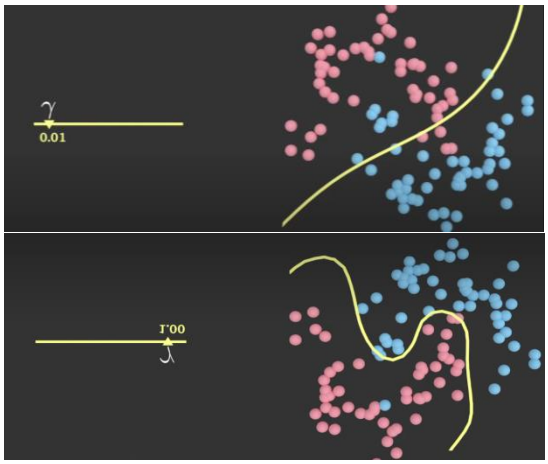

Figure 9: Representation of two decision boundaries with different values of Gamma [16]

The parameters C and Class_weight were also tuned with value of C optimal at 1 and class weight 0:0.5.3, 1:0.55.

V. Results

This section covers the results obtained from the experiment, a more detailed analysis of the results is covered in the discussion section.

Table 1: Results of the Experiment

Models			Evaluation Metrics			
			Precision	***Recall***	***F1 score***	***F1 Score***
Decision Tree	Before tunning	0	0.96	0.73	0.83	0.78
		1	0.22	0.73	0.34	
	After tunning	0	0.96	0.72	0.83	0.78
		1	0.22	0.74	0.34	
Logistic Regression	Before tunning	0	0.98	0.71	0.82	0.78
		1	0.23	0.83	0.36	
	After tunning	0	0.95	0.87	0.91	0.86
		1	0.32	0.57	0.41	
SVM	Before tunning	0	0.97	0.72	0.83	0.78
		1	0.23	0.82	0.36	
	After tunning	0	0.97	0.76	0.85	0.81
		1	0.25	0.74	0.37	
Decision Tree	Original Data	0	0.92	0.91	0.92	0.85
		1	0.24	0.28	0.26	

VI. Discussion

As shown in the previous section all the measures taken to balance the labels slightly improved the results. The resampling slightly decreased the model's ability to predict the majority label due to the information loss, but it also improved the model's predictions for the label 1.

Regarding the hyperparameter tunning it clearly makes a difference for logistic regression but not for decision tree and SVM. The results of logistic regression clearly improved after the tunning, the model is now better at predicting the minority label than without the tunning. The decision tree algorithm is a simple algorithm therefor the hyperparameter tunning didn't improve the results, the resampling however did. The SVM hyperparameter tunning posed many issues, the first one is because it is a very complex model it takes a lot of time to run so it becomes harder to test with this algorithm. Even after the size of the data was significantly reduced to improve the execution speed the model still took too long to run.

Overall logistic Regression is the best model, it's able to predict label 1 with 41% accuracy more than any other model and predicts label 0 with 91% accuracy also more than any other model.

VII. Conclusion

The results reflect the issue that is working with imbalanced data, in my opinion all the techniques that we could have employed to solve this issue were implemented, namely resampling techniques and weighted algorithms. Despite the implementation of these solutions the models still can't predict both labels with similar accuracy which leads us to think that is an issue with the data. The data was collected through a questionnaire over the phone and this kind of data can't be trusted to make medical decisions such as this one.

There is a need for more research in this area. A lack of medical staff is present in hospitals around the world and machine learning can be a solution for this problem, not to replace doctors but at least as a primary observation followed by the doctors approval. More researchers should work on this area to improve the current state of research. Also more effective data collection need to be in place to allow these researchers to achieve effective results, or get real patient data available publicly. Without quality data publicly available the field can't move forward.

VIII. Appendix

All the code used in this Paper is available in this git hub repository: https://github.coventry.ac.uk/lopesoll/7072-Machine-learning

IX. References

[1] Ganegoda, G. (2018). Secondary Prevention of Cardiovascular Diseases and Application of Technology for Early Diagnosis. https://www.researchgate.net/publication/325033300_Secondary_Prevention_of_Cardiovascular_Diseases_and_Application_of_Technology_for_Early_Diagnosis. Hindawi BioMed Research International, Volume 2018, https://doi.org/10.1155/2018/5767864

[2] McGill, H. (2008), "Preventing heart disease in the 21st century implications of the pathobiological determinants of atherosclerosis in youth (PDAY) study," Circulation, vol. 117, no. 9, (pg. 1216–1227) J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] Kozan, M. (2021). https://medium.com/@metehankozan/supervised-and-unsupervised-learning-an-intuitive-approach-cd8f8f64b644)

[4] Mitchell, T. (1997). Machine Learning. McGraw Hill Companies (pg. 52)

[5] Saini, A. (2021). Decision Tree Algorithm – A Complete Guide. https://www.analyticsvidhya.com/blog/2021/08/decision-tree-algorithm/

[6] Verma, Y. (2021). A Complete Guide to Decision Tree Split using Information Gain. (https://analyticsindiamag.com/a-complete-guide-to-decision-tree-split-using-information-gain/)

[7] Pant, A. (2019). Introduction to Logistic Regression. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

[8] Lee, W. (2019). Python Machine Learning. John Wiley & Sons. (pg. 180)

[9] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[10] Mwanthi, D. (2022). Getting started with Recursive Feature Elimination algorithm in Machine Learning

https://www.section.io/engineering-education/recursive-feature-elimination/#:~:text=the%20RFE%20algorithm.-,Implementing%20RFE%20algorithm,data%20into%20a%20decision%20tree.&text=The%20output%20above%20shows%20that%20the%20optimal%20number%20of%20features%20is%20t

[11] *Under-sampling*. Imbalanced-learn.org. (2022). Retrieved December 14, 2022, from https://imbalanced-learn.org/stable/under_sampling.html#edited-nearest-neighbors

[12] *Sklearn.preprocessing.StandardScaler*. Sklearn. (2022). Retrieved December 12, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html#sklearn.preprocessing.StandardScaler

[13] *1.13. feature selection*. scikit-learn.org. (2022). Retrieved December 14, 2022, from https://scikit-learn.org/stable/modules/feature_selection.html#rfe

[14] *Sklearn.GRIDSEARCHCV*. scikit-learn.org. (2022). Retrieved December 14, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV

[15] Narkhede, S. (2021, June 15). *Understanding confusion matrix*. Medium. Retrieved December 14, 2022, from https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[16] Johnson, J. (2020, July 22). *Precision, recall & confusion matrices in Machine Learning*. BMC Blogs. Retrieved December 14, 2022, from https://www.bmc.com/blogs/confusion-precision-recall/

[17] *8. common pitfalls and recommended practices#*. imbalanced-learn.org. (2022). Retrieved December 15, 2022, from https://imbalanced-learn.org/stable/common_pitfalls.html

[18] Sklearn.logisticregression. scikit-learn.org. (2022). Retrieved December 15, 2022, from https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[19] YouTube. (2022). *The Kernel Trick in Support Vector Machine (SVM)*. Retrieved December 14, 2022, from https://www.youtube.com/watch?v=Q7vT0--5VII.