# Comparison of Machine Learning techniques for Classification Problems


## Abstract

The study investigates the performance of five supervised learning algorithms on two classification datasets with distinct properties. The algorithms analyzed are Logistic Regression CV, Neural Networks, Boosted Trees, Support Vector Machines, and k-Nearest Neighbors. The difference between the classifiers is observed and analyzed to gain insights into their relative strengths and limitations.

In this project, we adopted some machine learning techniques and one of the deep learning custom designed neural networks. We compared the performances of the previously mentioned algorithms for classifying the heart disease prediction in the patients. Similarly, the second dataset we are using is about hotel reservation for the booking purposes.


### INTRODUCTION

A class of machine learning technique known as supervised learning uses labeled data to train predictions. The algorithms learn to predict the presence or absence of heart disease based on the clinical parameters included in the dataset when applied to the Cleveland heart disease dataset. A supervised learning algorithm's effectiveness can be assessed using a number of metrics, including accuracy, precision, recall, and F1 score. The algorithms will develop a mapping from the clinical parameters to the binary label indicating the presence or absence of heart disease when they are applied to the dataset. Finding the best mapping, or model, that performs well on unseen data—data that wasn't used during training—is the objective. On the dataset, some algorithms might outperform others because they have various strengths and drawbacks. The performance of several algorithms can be compared in order to learn more about the relative advantages and disadvantages of each algorithm and to identify which method is most appropriate for this particular task.

A wealth of data on client hotel reservations is accessible in the hotel booking dataset on Kaggle. It includes a number of attributes, including the booking date, arrival date, number of adults and children, country of origin, and cancellations of bookings. The job of predicting whether a booking will be canceled or not based on the provided attributes presents an interesting challenge for supervised learning systems in this dataset. Predicting one of two probable outcomes is the objective of this kind of task, which is often referred to as a binary classification problem.

The hotel reservation dataset presents a special chance to deploy supervised learning algorithms and assess their effectiveness on a practical issue. It is possible to identify patterns and links

between booking features and cancellations by training machine learning models on the dataset; these models can then be used to forecast outcomes for future data that has not yet been collected. Hotel management that want to anticipate cancellations and take preventative measures to lessen their impact on revenue may find this to be helpful. The hotel reservation dataset will be utilized in this study as the second classification dataset to examine the effectiveness of different supervised learning methods.

## Data Exploration

A frequently used dataset in the fields of machine learning and medical research is the Cleveland Heart Disease dataset, which is accessible on Kaggle. With only 303 observations of patients with heart disease and some of their demographic and medical characteristics, it is a smaller dataset than the Heart Disease dataset I discussed before.

The features in this dataset include Age, Sex, Chest pain type, Resting blood pressure, Serum cholesterol levels, Fasting blood sugar levels, Resting electrocardiographic results, Maximum heart rate achieved, Exercise induced angina, ST depression induced by exercise relative to rest, The slope of the peak exercise ST segment, Number of major vessels colored by fluoroscopy, Thalassemia.The target variable in this dataset is binary, indicating whether or not a patient has heart disease, and is represented by the value of 1 or 0. Below is the sample data from the heart dataset:

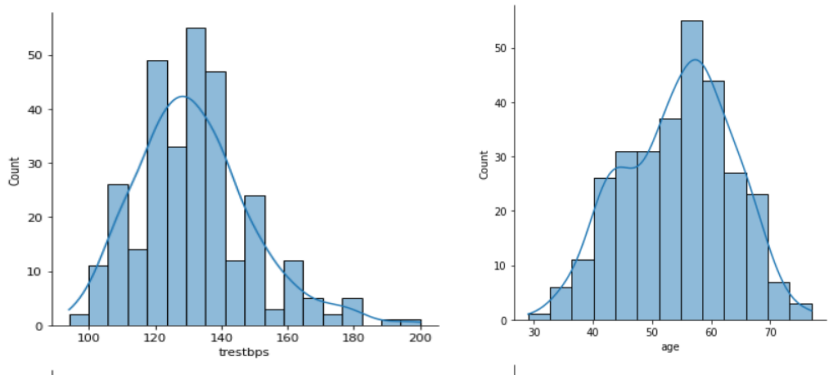
| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 63 | 1 | 3 | 145 | 233 | 1 | 0 | 150 | 0 | 2.3 | 0 | 0 | 1 | 1 |
| 1 | 37 | 1 | 2 | 130 | 250 | 0 | 1 | 187 | 0 | 3.5 | 0 | 0 | 2 | 1 |
| 2 | 41 | 0 | 1 | 130 | 204 | 0 | 0 | 172 | 0 | 1.4 | 2 | 0 | 2 | 1 |
| 3 | 56 | 1 | 1 | 120 | 236 | 0 | 1 | 178 | 0 | 0.8 | 2 | 0 | 2 | 1 |
| 4 | 57 | 0 | 0 | 120 | 354 | 0 | 1 | 163 | 1 | 0.6 | 2 | 0 | 2 | 1 |

Similarly, we have the second dataset which is about hotel reservation. Customers' hotel reservations are detailed in the Hotel Reservation dataset, which is accessible on Kaggle. It serves as an example of how machine learning algorithms can be used in the hospitality and tourism industries. Below is the sample data from the hotel reservation dataset:

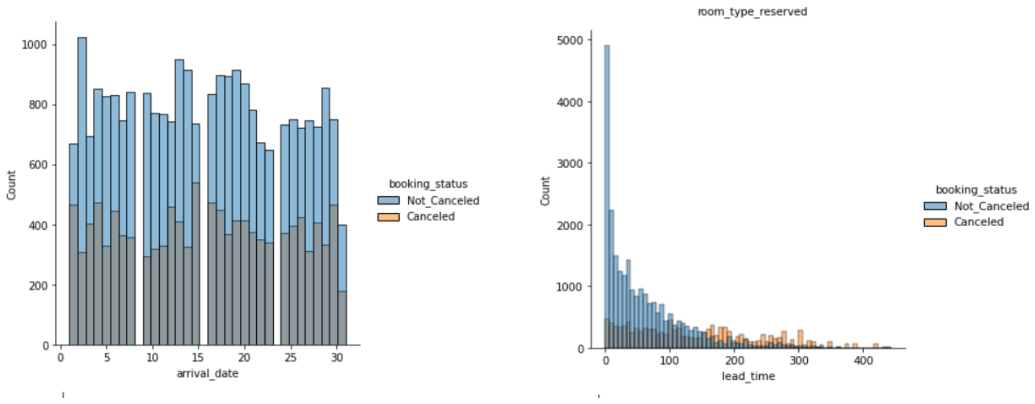
| | Booking_ID | no_of_adults | no_of_children | no_of_weekend_nights | no_of_week_nights | type_of_meal_plan | required_car_parking_space |
|---|---|---|---|---|---|---|---|
| 0 | INN00001 | 2 | 0 | 1 | 2 | Meal Plan 1 | 0 |
| 1 | INN00002 | 2 | 0 | 2 | 3 | Not Selected | 0 |
| 2 | INN00003 | 1 | 0 | 2 | 1 | Meal Plan 1 | 0 |
| 3 | INN00004 | 2 | 0 | 0 | 2 | Meal Plan 1 | 0 |
| 4 | INN00005 | 2 | 0 | 1 | 1 | Not Selected | 0 |

The features in this dataset typically include Hotel type , Check-in date, Check-out date, Arrival date year, Stays in weekend nights,Adults,, Country, Market segment, Distributed channel , Is repeated guest, Previous cancellations, Previous bookings not canceled, Reserved room type,etc. The target variable in this dataset is binary and indicates whether a customer has canceled their booking or not, represented by the value of 1 or 0.

Now, we will analyze and explore our first dataset from a univariate, bivariate and multivariate perspective to describe it graphically. It will give us a better intuition of the data.

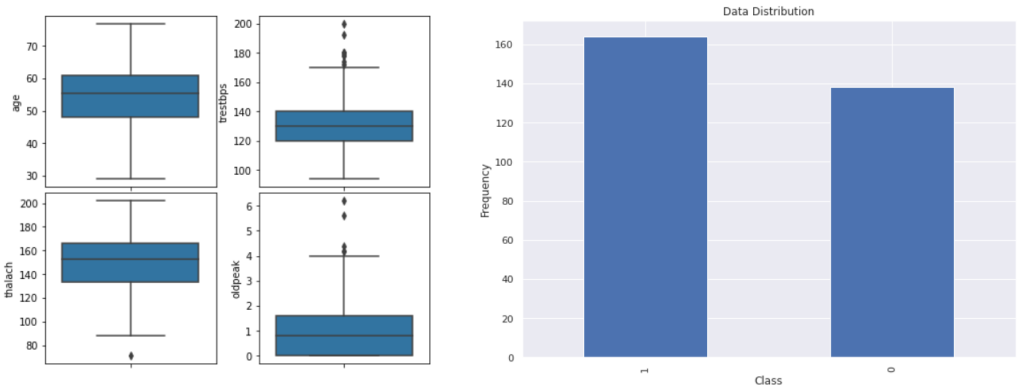



Now, we will analyze and explore our second dataset from a univariate, bivariate and multivariate perspective to describe it graphically. It will give us a better intuition of the data.

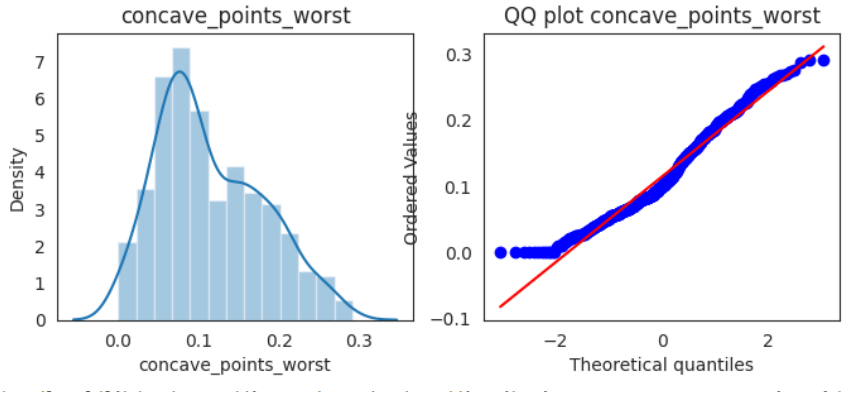



Above figures show the univariate analysis of the two features segmented upon target variable diagnosis. Blue bars show one type of data for one disease while orange bar plots show the plots for the other class.

Now, we will see if there are any outliers in the first dataset named as heart prediction dataset.



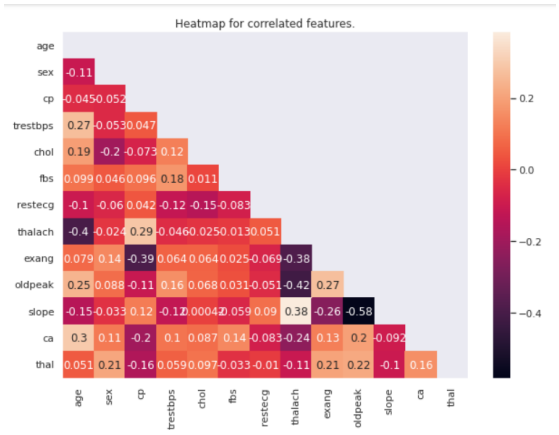

Similarly, bivariate analysis of some features of the second dataset is shown below in figure that either the data has features which fit linearity using QQ prob plot:

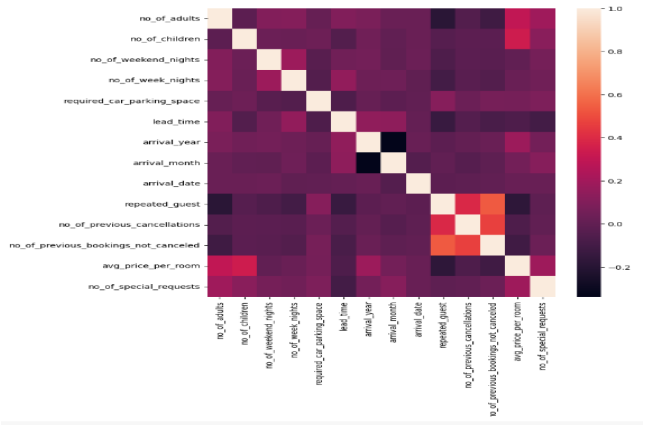



Above plot shows that the feature has skewed a bit from the left side. Hence, our feature has linear relation and the distribution is of gaussian form.

On the left side, we are observing the feature distribution and data spread is either normalized or gaussian type. While , on the right side, we are checking the linearity of the data. Below we are checking the multi covariance of both datasets.

It offers a visual evaluation of whether or not the sample data exhibits a normal distribution. In other words, it determines if the sample data's distribution resembles a normal distribution. In statistical analysis and machine learning, QQ plots are frequently used to identify potential problems with data distribution and to assess the reliability of statistical tests.




## Data Preparation

First, we loaded the dataset and checked its stats and info. After confirming the number of continuous valued features number and categorical features, we handled the missing values but as there are not any missing values. Hence , we head on. Then, we handled the categorical features in which the target variable ‘target’ was included and we mapped its binary values with 0 and 1. Then, we handled the numerical features in which we removed the outliers using the z-score method.

Then , we splitted the dataset into X and Y variables. After that we scaled the X features columns using the StandardScaler. Then, we splitted the X,y variables into training and testing sets for the further biased evaluation. After the whole data preparation, we have the data which would look like :

| | age | sex | cp | trestbps | chol | fbs | restecg | thalach | exang | oldpeak | slope | ca | thal | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 | 303.000000 |
| mean | 54.366337 | 0.683168 | 0.966997 | 131.623762 | 246.264026 | 0.148515 | 0.528053 | 149.646865 | 0.326733 | 1.039604 | 1.399340 | 0.729373 | 2.313531 | 0.544554 |
| std | 9.082101 | 0.466011 | 1.032052 | 17.538143 | 51.830751 | 0.356198 | 0.525860 | 22.905161 | 0.469794 | 1.161075 | 0.616226 | 1.022606 | 0.612277 | 0.498835 |
| min | 29.000000 | 0.000000 | 0.000000 | 94.000000 | 126.000000 | 0.000000 | 0.000000 | 71.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| 25% | 47.500000 | 0.000000 | 0.000000 | 120.000000 | 211.000000 | 0.000000 | 0.000000 | 133.500000 | 0.000000 | 0.000000 | 1.000000 | 0.000000 | 2.000000 | 0.000000 |
| 50% | 55.000000 | 1.000000 | 1.000000 | 130.000000 | 240.000000 | 0.000000 | 1.000000 | 153.000000 | 0.000000 | 0.800000 | 1.000000 | 0.000000 | 2.000000 | 1.000000 |
| 75% | 61.000000 | 1.000000 | 2.000000 | 140.000000 | 274.500000 | 0.000000 | 1.000000 | 166.000000 | 1.000000 | 1.600000 | 2.000000 | 1.000000 | 3.000000 | 1.000000 |
| max | 77.000000 | 1.000000 | 3.000000 | 200.000000 | 564.000000 | 1.000000 | 2.000000 | 202.000000 | 1.000000 | 6.200000 | 2.000000 | 4.000000 | 3.000000 | 1.000000 |

| | no_of_adults | no_of_children | no_of_weekend_nights | no_of_week_nights | type_of_meal_plan | required_car_parking_space | room_type_reserved |
|---|---|---|---|---|---|---|---|
| count | 25392.0 | 25392.0 | 25392.0 | 25392.0 | 25392.0 | 25392.0 | 25392.0 |
| mean | 1.8 | 0.1 | 0.8 | 2.2 | 0.5 | 0.0 | 0.7 |
| std | 0.5 | 0.3 | 0.9 | 1.3 | 1.0 | 0.1 | 1.4 |
| min | 0.3 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 | 0.0 |
| 25% | 2.0 | 0.0 | 0.0 | 1.0 | 0.0 | 0.0 | 0.0 |
| 50% | 2.0 | 0.0 | 1.0 | 2.0 | 0.0 | 0.0 | 0.0 |
| 75% | 2.0 | 0.0 | 2.0 | 3.0 | 0.0 | 0.0 | 0.0 |
| max | 3.4 | 1.3 | 3.4 | 6.4 | 3.0 | 0.6 | 4.9 |

## Experimental Procedure

We have used the five classification algorithms which perform best of all. These models include Logistic Regression CV,  Support Vector Machines, KNN, Boosted Trees and Artificial Neural Network. First four algorithms belong to machine learning techniques and the last one is deep neural networks . logistic regression classifies the entities using sigmoid activation function with the gradient descent algorithm. While the support vector machine uses laplacian arguments to find the best possible solution. Similarly, the deep neural network uses the adam optimizer.

These are the results of the first dataset on various models:

- **Decision Tree**

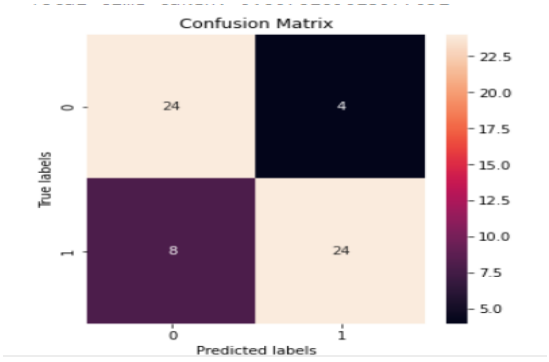
  The feature that best distinguishes the target variable is used by the decision tree algorithm to iteratively split the data into smaller and smaller subsets. The method divides the data into two or more subgroups based on the values of the feature that reduces impurity the most (as determined by metrics like entropy or the Gini index) at each phase. The accuracy we got from this model is 75% on both sets and precision we got 84%. We can see the learning curves and confusion matrix:

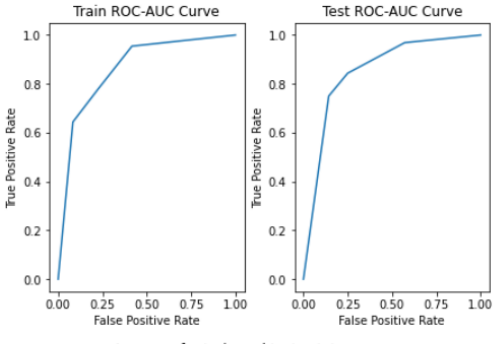
```
Mean F score on train set is: 0.766118802262195
Mean F score on test set is: 0.7386330144492069

Mean Precision on train set is: 0.8422219995992787
Mean Precision on test set is: 0.8422219995992787

Mean Recall on train set is: 0.7043603978718483
Mean Recall on test set is: 0.6651515151515152

Mean Train ROC is: 0.860326887037359
Mean Test ROC is: 0.8155138938993106
Mean Train loss is: 1340.1045241225338
Mean Test loss is: 379.92654034401744
Total time taken: 0.6076169013977051
```

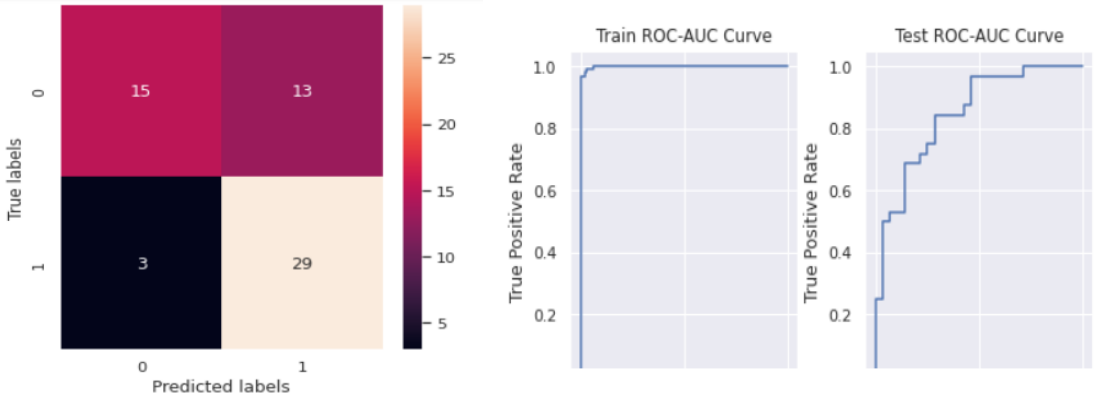



- **XGBoost**

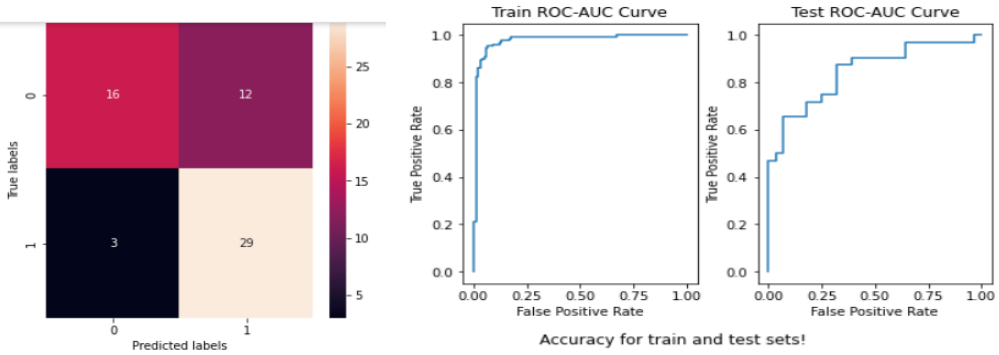
  The gradient boosting algorithm has been enhanced in XGBoost (eXtreme Gradient Boosting). It is a potent machine learning technique that is applied to both classification and regression issues. Simply put, XGBoost creates an ensemble of decision trees, where each tree is fitted using the residuals (or errors) of the prior tree. The weighted sum of the

predictions made by each tree in the ensemble is the final forecast. The accuracy we got from this model is 98% on training and 80% on testing sets and precision we got 84%. We can see the learning curves and confusion matrix:



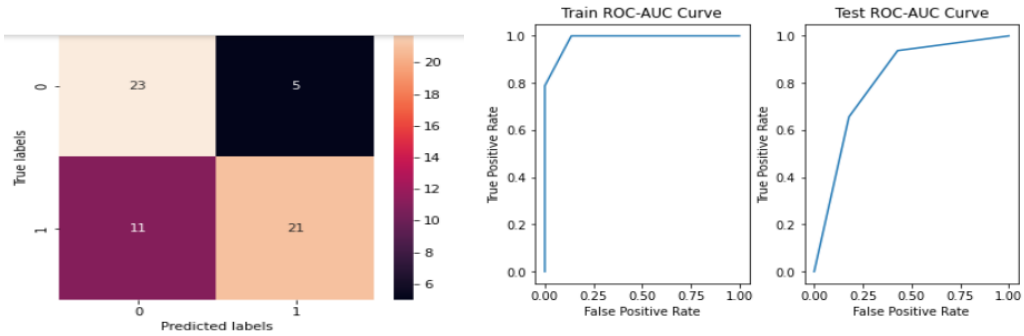

- **SVM**

SVM determines the hyperplane with the largest margin in a two-class classification issue, where margin is the separation between the nearest data points for each class and the hyperplane. The hyperplane is defined using support vectors, which are the nearest data points. If the data is not linearly separable, SVM can still be used by transforming the data into a higher-dimensional space where a linear boundary can be found. The accuracy we got from this model is 91% on training and 82% on testing sets and precision we got 82%. We can see the learning curves and confusion matrix:



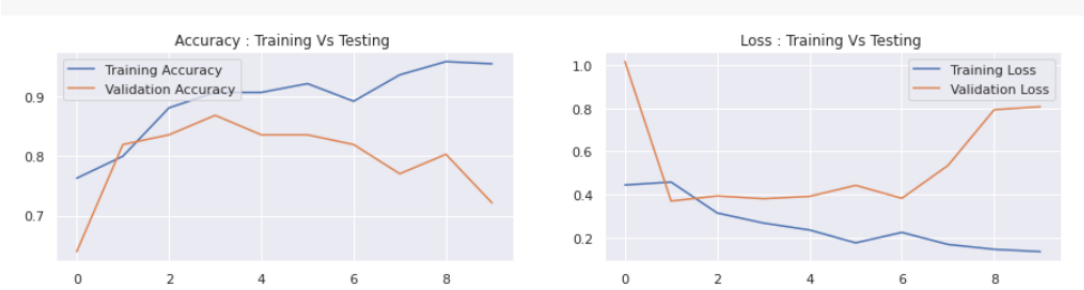

- **KNN**

By calculating the number of data points from each class among the K closest neighbors and choosing the class with the highest count, KNN finds the class label for a new observation in a classification issue. By averaging the target variable among the K closest neighbors, KNN may forecast the target variable for a new observation in a regression issue. The accuracy we got from this model is 89% on training and 80% on testing sets and precision we got 81%. We can see the learning curves and confusion matrix:




- **Deep neural Network**

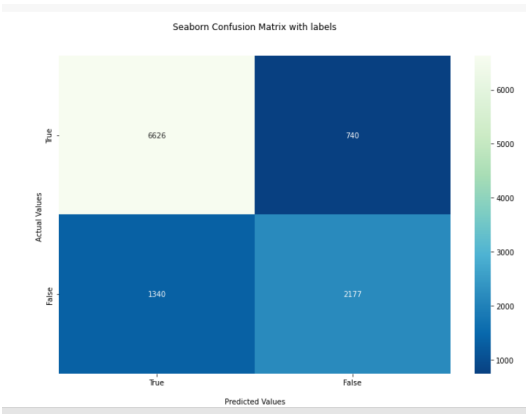
  The artificial neural network is a deep learning architecture which uses the adam optimizer which combines the properties of all optimizers. We may have to tweak the learning rate to optimize the performance of the algorithm. The accuracy we got from this model is 92% on training and 85% on testing sets and precision we got 84%. We can see the learning curves and confusion matrix:

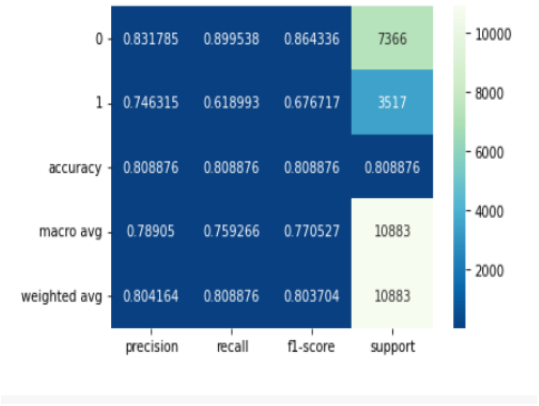



Now, for the second dataset, we have the following results:

- **Logistic Regression CV**

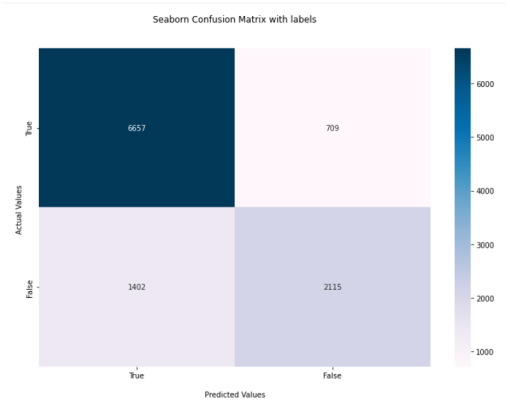
Logistic Regression CV is a version of simple logistic regression which applies the sigmoid activation function at output and uses the same gradient descent algorithm as in the linear regression. Further, we have to tweak some parameters to get the optimal accuracy.The confusion matrix and classification report shows the performance of this model. Our model has given the accuracy of 80% on the testing set while the precision and recall score is 79% and 81% respectively

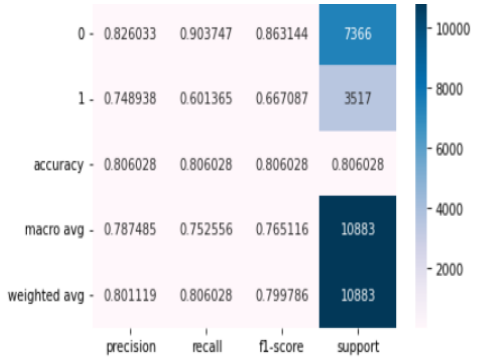


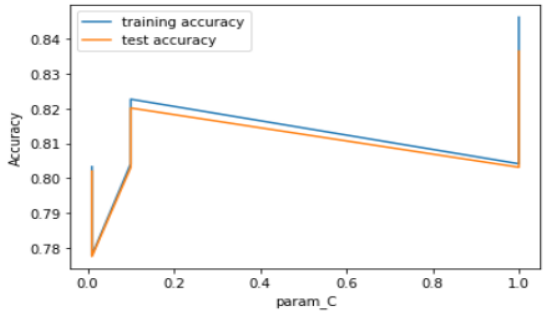




- **Support Vector Machines**

SVM determines the hyperplane with the largest margin in a two-class classification issue, where margin is the separation between the nearest data points for each class and the hyperplane. The hyperplane is defined using support vectors, which are the nearest data points. If the data is not linearly separable, SVM can still be used by transforming the data into a higher-dimensional space where a linear boundary can be found. The accuracy we got from this model is 82.6% on training and 80% on testing sets and precision we got 78%%. We can see the learning curves and confusion matrix:

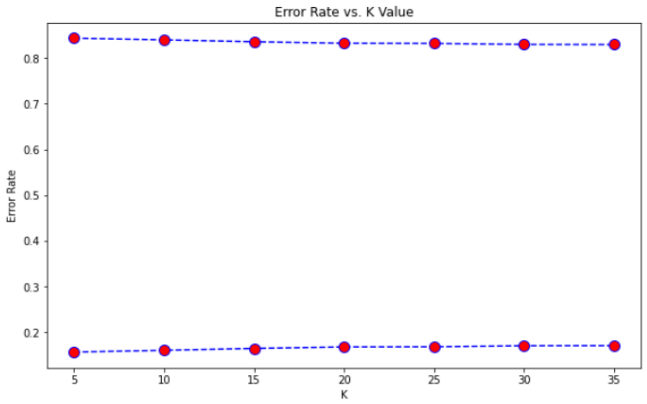


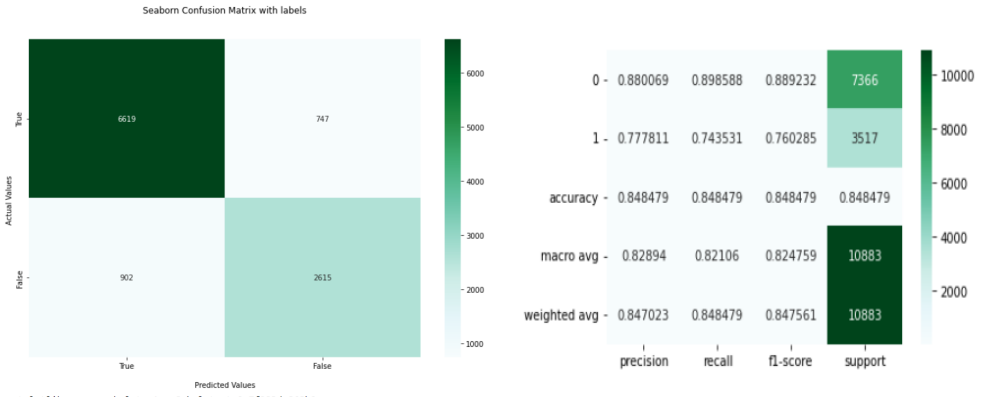



- **KNN**

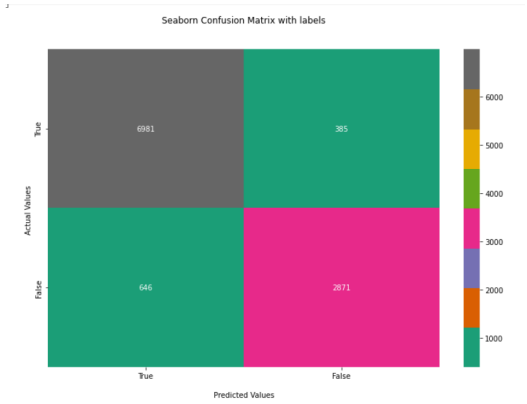
  By calculating the number of data points from each class among the K closest neighbors and choosing the class with the highest count, KNN finds the class label for a new observation in a classification issue. By averaging the target variable among the K closest neighbors, KNN may forecast the target variable for a new observation in a regression issue. The accuracy we got from this model is 89% on training and 85% on testing sets and precision we got 84%. We can see the learning curves and confusion matrix:

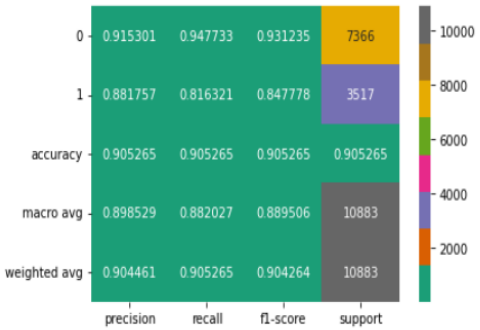



- **Random Forest**

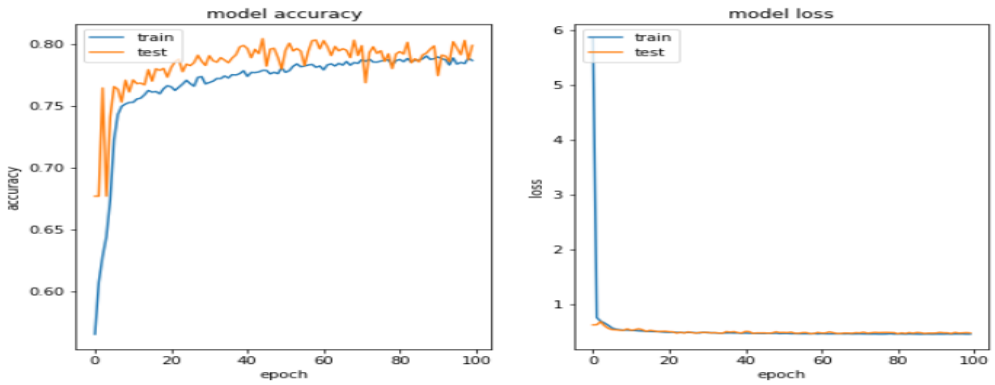
  Random Forest is an ensemble machine learning algorithm that builds multiple decision trees and combines their predictions to obtain a more accurate and stable result. The algorithm works by creating multiple decision trees from bootstrapped samples of the training data, and then aggregating the predictions of the individual trees to produce the final prediction. The accuracy we got from this model is 90% on training and 90.6% on testing sets and precision we got 89.8%. We can see the learning curves and confusion matrix:




- **Deep Neural Network**

The artificial neural network is a deep learning architecture which uses the adam optimizer which combines the properties of all optimizers. We may have to tweak the learning rate to optimize the performance of the algorithm. The accuracy we got from this model is 81% on training and 80% on testing sets and precision we got 80%. We can see the learning curves and confusion matrix.The performance of deep learning algorithm artificial neural networks also performs much better without the tuning of the models. Below, we can see the learning curves of the model and how it avoids overfitting on the test set.




## Conclusion

We tuned all the models by their random hyper parameters to see their full performance and we got that the deep neural network outperformed all the algorithms for the first dataset of heart disease prediction and it gave 91% accuracy while the random forest outperformed all the algorithms ,and it maintained 90% accuracy without overfitting, which include KNearest Neighbors , SVM , Deep Neural Network. But the thing is we have to keep two things in mind. First, is the tweaking of parameters to get the best accuracy . Second, we have to avoid overfitting the model.

# References

[1] Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547-553.

[2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.

[3] https://www.kaggle.com/datasets/ahsan81/hotel-reservations-classification-dataset

[4] Mohan S., Thirumalai C. and Srivastava G. 2019 Effective heart disease prediction using hybrid machine learning techniques *IEEE Access* 7 81542-81554.

[5] Ramalingam V. V., Dandapath A. and Raja M. K. 2018 Heart disease prediction using machine learning techniques: a survey *International Journal of Engineering & Technology* 7 684-687