SVM & Random Forest Classifier

Evaluating the performance of a simple learning system on a real-world dataset.

Introduction

In this project we are using real life lung cancer data and implementing it on two machine learning models “Support Vector Machine” and “Random Forest”. We will be doing pre-processing of data and then making data into numerical form which will be taken by these models. After that we will be evaluating the model by using five evaluation metrics, including accuracy, precision, recall, f1 score, and AUC (area under curve of receiver operating characteristics (ROC) curve).

Pre-processing

First we will be understanding data and seeing if data balances or not. We will import important packages first and after that we will see the top 5 entries in data. After that we will see how many columns are there in the dataset and see some statistics about it. We will also be confirming that the data doesn't contain any null value. After that we will be plotting a count plot on the class column.

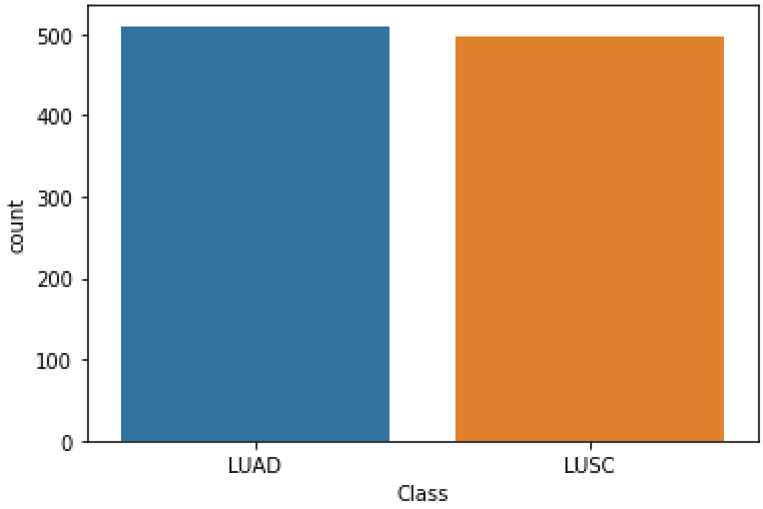


Figure1. Count plot of Class

In this Chart we can clearly see that both our classes are balanced and have equal amounts of entries. After that we see that our data is balanced we will move towards splitting data and assigning X and Y, and splitting them into training and testing data. We will use a train test split from sklearn and take 20 percent data into testing. After all of this our data is ready to go into machine learning models.

SUPPORT VECTOR MACHINE

(a) What it does:

Support Vector Machine is a classical machine learning algorithm that falls under the category of supervised learning. SVMs are used for both classification and regression tasks.

(b) How it does:

Support Vector Machines (SVM) excel at categorizing data points, even when linear separation seems challenging. They achieve this by mapping the data into a higher-dimensional feature space. SVMs are versatile, applied to both classification and regression tasks, and are particularly valuable in scenarios with intricate decision boundaries. In the transformed space, each data point becomes a point in a layered structure, where the significance of each feature dictates its position. Support vectors, representing critical instances, play a vital role in determining the optimal hyperplane or line that effectively separates different classes. SVMs are robust classifiers, leveraging individualized kernel functions to handle non-linear relationships, making them powerful tools for diverse machine learning applications, especially in contexts where conventional linear separation methods fall short.

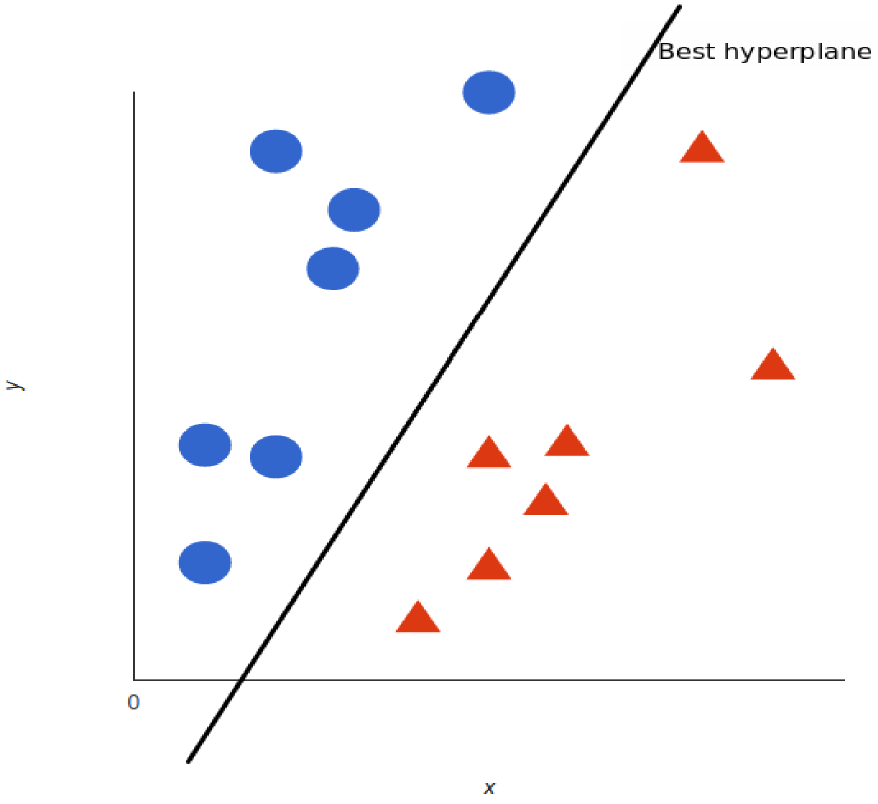


Fig 2: Working of SVM

In this figure we can clearly see how the Support vector machine works and classify two classes. First we will fit the model into our training data and after that do prediction on our test data. The result was described below.

(c) Application:

Facial expression, texture classification and speech recognition are some of the applications of Support Vector Machine.

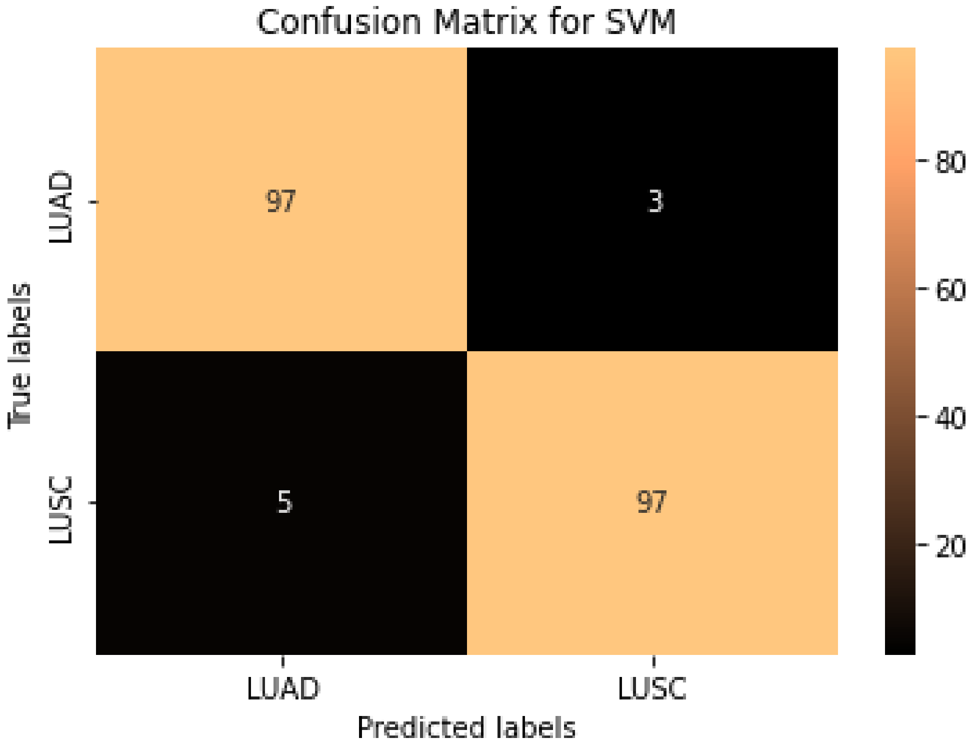
(A) RESULTS FOR SUPPORT VECTOR MACHINE

(B) Table 1: Classification Report of SVM

	Precision	**Recall**	**F1-score**	**Support**
0	0.95	0.97	0.96	100
1	0.97	0.95	0.96	102
Accuracy			0.96	202
macro avg	0.96	0.96	0.96	202
weighted avg	0.96	0.96	0.96	202

(C) OBSERVATION of Table 1:

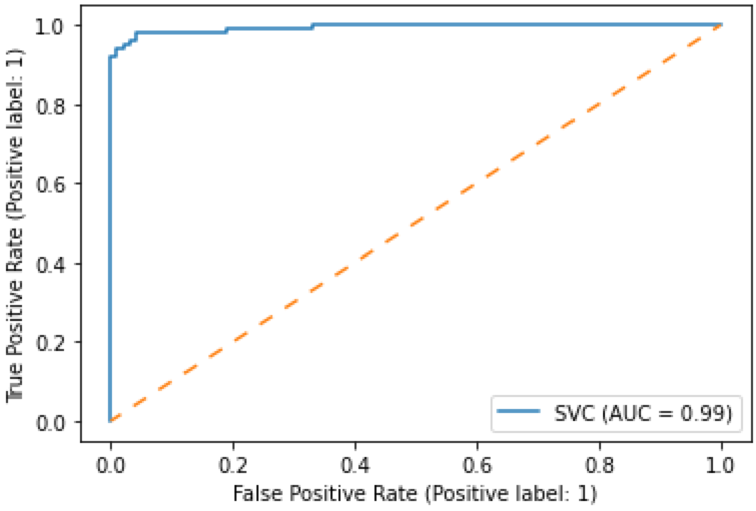
This table shows that Support Vector Machine is giving 0.96 accuracy and precision of 0.95 on 0 class (LUAD) and 0.97 on 1 class(LUSC). We can see recall and f1-score in the above table.



(B) Fig 3: Confusion matrix of SVM

(C) OBSERVATION of Fig 3:

This confusion matrix explains the performance of our model by seeing the right answer given by model and wrong answer. In this confusion matrix we can see the model is giving only 8 wrong answers.



(B) Fig 4: Roc Curve of SVM

(C) OBSERVATION of Fig 4: By looking at this ROC curve of SVM we can say that the model has 0.99 ability to separate between two classes

(D) CONCLUSION:

To find a hyperplane in an N-dimensional space that distinctly classifies the data points. The best method to evaluate your classifier is to train the SVM algorithm with 80% of your training data and 20% to test your classifier.

RANDOM FOREST CLASSIFIER

(a)What it does:

Supervised machine learning algorithms like random forest are commonly used for both classification and regression tasks. In the case of classification, random forest constructs multiple decision trees on different subsets of the training data and combines their predictions through a majority vote, determining the class with the most votes as the final prediction. For regression tasks, random forest similarly builds decision trees on various samples of the training data, but instead of a majority vote, it takes the average of the predictions from individual trees to provide the final regression output. This ensemble approach helps random forest models to generalize well and improve overall predictive performance.

(b) How it does:

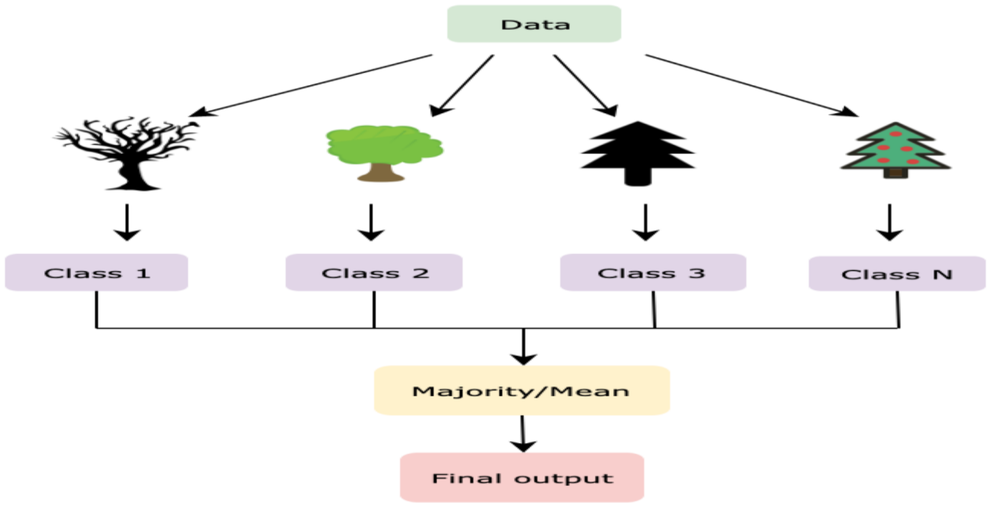


Fig 5: Random Forest working

A supervised learning algorithm example could be random forest. The "forest" it makes is an outfit of choice trees, which are many times prepared utilising the "bagging" approach. The bagging approach depends on the reason that consolidating learning models works on the last result.

Irregular backwoods have practically indistinguishable hyper parameters as choice trees and packing classifiers. Luckily, there is a compelling reason to join a choice tree with a packing classifier in light of the fact that the irregular woodland classifier-class might be utilised all things being equal. You may likewise utilise arbitrary backwoods to deal with relapse occupations by utilising the calculation's regression.

Irregular backwoods give capriciousness to the model while the trees create. While parting a hub, it searches for the best component from an irregular gathering of qualities as opposed to the most fundamental element.

Subsequently, there is a more noteworthy assortment, which prompts a superior model. Subsequently, the procedure for parting a hub in some irregular backwoods thinks about only an irregular subset of the qualities. You might try and make trees more irregular by applying irregular limits for each element as opposed to searching for the most ideal edges (like a typical choice tree does).

(c) Application:

Some of the applications of Random Forest classifier are Radar, communication, Access control systems. It also used in industrial data acquisition system and wireless data terminals.

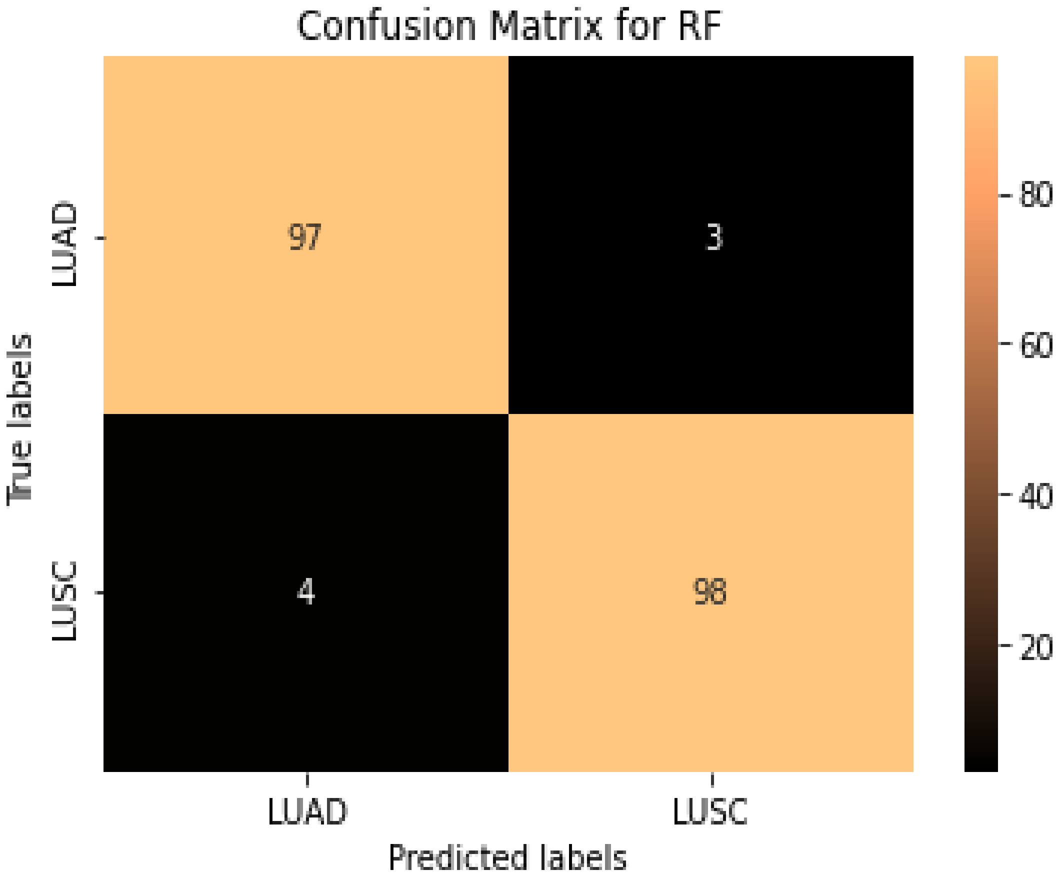
(A) RESULTS FOR RANDOM FOREST CLASSIFIER

(B) Table 2: Classification Report of Random Forest

	Precision	**Recall**	**F1-score**	**Support**
0	0.96	0.96	0.96	100
1	0.96	0.96	0.96	102
Accuracy			0.96	202
macro avg	0.96	0.96	0.96	202
weighted avg	0.96	0.96	0.96	202

(C) OBSERVATION of Table 2:

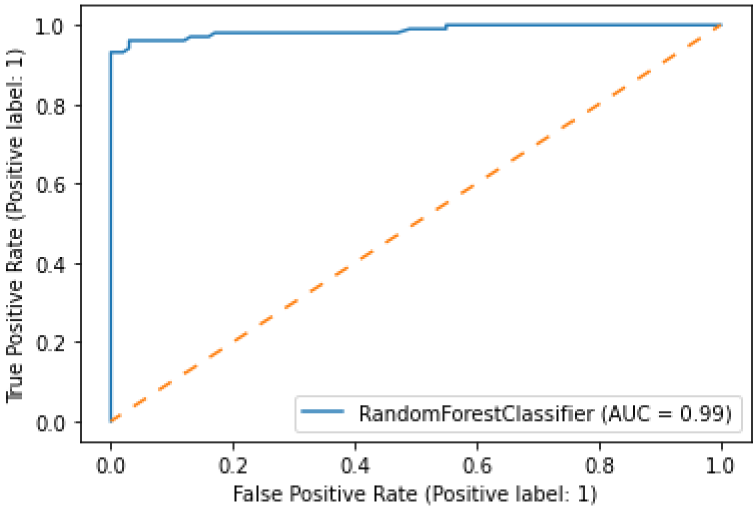
As we understand how it works we will import Random forest from sklearn and fit and predict our data same as SVM. This shows us that Random forest is giving 0.96 numbers in every field, which means it has accuracy, recall, precision and f1 score of 96 percent.



(B) Fig 6: Confusion matrix of Random Forest

(C) OBSERVATION of Fig 6:

This plot explains the performance of our model by seeing the right answer given by model and wrong answer. In this confusion matrix we can see the model is giving only 7 wrong answers which are more than the SVM model.



(B) Fig 7: Roc Curve of Random Forest

(C) OBSERVATION of Fig 7:

By looking at this plot we can say that the model has 0.99 ability to separate between two classes

Roc Curve Comparison of Random Forest & SVM

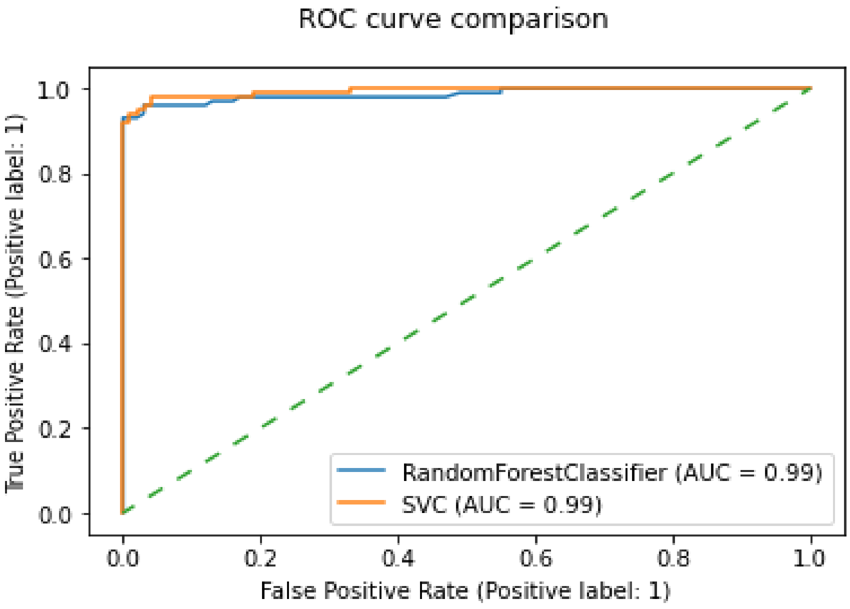


Fig 8: ROC Curve of Random Forest & SVM

OBSERVATION of Fig 8:

This plot explains as FPR is false positive rate and TPR is true positive rate thresholds are different probability cut-offs that separate the two classes. First, we are taking values from real y labels and predicted y of the model, these three values and after that we are just plotting these things on a graph. Taking TPR on y axis and FPR on x axis. As the score is close to 1 it means that model is working well and has the ability to do class action between two classes. By looking at this plot we can say that both of the models have 0.99 ability to separate between two classes. There is not much difference between both models ROC Curves and AUC scores.

CONCLUSION

As we have used two machine learning Model for lungs cancer prediction, we can see that both model gave similar accuracy and result on metrics of test data. However as Random Forest have constant 0.96 score and SVM have 0.95 score in precision and recall. I will be giving more preference to Random Forest as it gave a consistent score of 0.96 is better than variation in scores whereby in SVM we have some issue in precision of 0 and recall of 1, so I will be giving more preference to consistency.