Artificial Intelligence (CSE 422) Project Report

Topic : Lung Cancer Prediction using machine learning techniques

Group Member's Name and ID:

1) Readhwana Reaz Adrin - 19201034
2) Mahmuda Junainah - 19201060
3) Masiat Hasin Rodoshi - 19201089

Group No : 7

Lab section : 3

Introduction

Lung cancer occurs when the cells in the lungs divide in an uncontrolled manner and leads to the formation of a tumor in the lungs, which eventually leads to lung cancer. Lung cancer is one of the common causes of cancer death in men and women. There are multiple causes of lung cancer and among them some common causes of lung cancer are both active and passive smoking, air pollution, being exposed to certain chemicals such as arsenic, tar, beryllium, nickel etc, being exposed to radiation and having lung cancer genetically. The symptoms of lung cancer include having a cough which lasts for more than two-three weeks, breathlessness, chest infections, fatigue, ache in the chest while coughing or breathing etc. Nowadays lung cancer can be predicted with the help of artificial intelligence and machine learning techniques. Artificial intelligence deals with the algorithms or computer programs which can interpret the data to derive conclusions or predictions. On the other hand machine learning is the algorithm that learns how to evaluate and understand data by itself. These machine learning algorithms are trained by feeding in new data to the machine learning algorithms and this causes the algorithms to improve its capacity to learn and comprehend data. One of the advantages of using machine learning to predict lung cancer is that machine learning algorithms can recognize patterns that are difficult for the human eye or brain to recognize.

In this project we are going to discuss and compare the machine learning techniques that can be used to predict lung cancer and through the comparison of different types of machine learning techniques we are going to find out the most accurate machine learning technique which can predict lung cancer accurately. For this project we have used a dataset from the website *Kaggle* and at first we have pre processed the data to implement the machine learning models. The models used in this project are Logistic regression, K-Nearest Neighbor (KNN), Naive Bayes and Random forest. The data pre-processing techniques and the implementation of the models will be discussed in detail in this lab report and we will also discuss the outcomes generated through these models. At the end of this report the accuracy of these models will be compared. The aim of this project is to figure out the best machine learning model which can predict lung cancer and the target audience of this project is the lung cancer researchers, who are working to develop the machine learning techniques to predict lung cancer and lung cancer patients. Using machine learning techniques to predict lung cancer will help to detect lung cancer at an early stage and this will also cause the death rate due to lung cancer to decrease.

Methodology

In this section, we first analyzed the dataset and then preprocessed it before sending it to the model. After that we aimed to train the model on different machine learning algorithms such as- Naive Bayes, Random Forest, Logistic Regression and K - Nearest Neighbor (KNN).

Data set description

In order to demonstrate the performance of our machine learning model we used a publicly available supervised dataset from an online website (kaggle.com). The name of our dataset is "Lung Cancer". The efficiency of cancer prediction systems enables people to determine their cancer risk at a minimal cost and to make the best decisions possible in light of that clinical status. In order to obtain this classification performance, we divided the dataset into training and testing sets. The dataset contains a total of 16 attributes. The attributes contain a total of 284 instances. The attributes have information about 15 features and 1 label. Each feature and label contain information regarding their attribute. The attribute information are as follows:

1. Gender: M(male), F(female)
2. Age: Age of the patient
3. Smoking: YES=2 , NO=1.
4. Yellow fingers: YES=2 , NO=1.
5. Anxiety: YES=2 , NO=1.
6. Peer_pressure: YES=2 , NO=1.
7. Chronic Disease: YES=2 , NO=1.
8. Fatigue: YES=2 , NO=1.
9. Allergy: YES=2 , NO=1.
10. Wheezing: YES=2 , NO=1.
11. Alcohol: YES=2 , NO=1.
12. Coughing: YES=2 , NO=1.
13. Shortness of Breath: YES=2 , NO=1.
14. Swallowing Difficulty: YES=2 , NO=1.
15. Chest pain: YES=2 , NO=1.
16. Lung Cancer: YES , NO.

Data Preprocessing techniques

Data preprocessing is a data mining approach that turns unstructured data into something that can be interpreted. Raw data (data from the real world) is always suboptimal and cannot be passed via a model. That can result in certain inaccuracies. Because of this, preprocessing data is necessary before sending it via a model. The steps we used in data preprocessing for this particular project are-

1. **Import libraries:** We are using Pandas and Numpy as our main libraries. Pandas is used for data manipulation and data analysis and Numpy is a fundamental Python package for scientific computing. We are also using Matplotlib and Seaborn for the visualization and Scikit-learn libraries for data preprocessing techniques and algorithms.

2. **Read data:** Here, we are reading the data in the CSV file using pandas.

3. **Get to know the data using data visualization and other techniques:** In order to manipulate the data, we used the data process Shape. The Shape process returns us the number of rows and columns in our dataset. For our dataset, the output is (309, 16).

4. **Checking for missing values:** At first, we looked for the number of null values in the dataset which returned us three features containing null values. Instead of dropping those values, we decided to use a missing values handling technique called “Imputing Missing Values”. In this technique, the null values in a particular feature get replaced by the mean value of that column. Thus, the dataset gets rid of null values.

5. **Checking for duplicate values:** Then, we checked for duplicate values (rows) in the dataset and dropped it which gave us a total of 283 rows and 16 columns.

6. **Encoding Categorical Features:** In this preprocessing method, we look for the attributes containing categorical values (text values). Then we transform it into an integer value using Label Encoder with the help of the fit_transform method. Instead of using fitand transform separately, we used the fit_transform method so that the model efficiency does not decrease.

7. **Feature selection:** In this process, we calculate the correlation value of each attribute with one another. The highest correlation value is 1 which is between the same attribute such as - GENDER and GENDER. Then we generate a heatmap depending on these correlation values using Seaborn. A correlation heatmap is a heatmap that portrays a two-dimensional correlation matrix between two distinct dimensions, with colored cells representing data from a monochromatic scale. The first dimension's values are displayed as the table's rows, while the second dimension's values are displayed as columns. The ultimate goal of this process is to drop features which are less important or redundant compared to the label. But In our project, we decided not to drop any feature as our dataset is already really small containing only 15 features.

8. **Data splitting:** In this step, we are splitting the feature array and label array keeping 80% for the training sets.

9. **Feature scaling:** A machine learning model tends to give more priorities to the features containing higher values compared to the one containing lower values. This ultimately results in a lower accuracy rate. In order to avoid this tendency, we used feature scaling. We used the MinMaxScaler in order to scale the data and range it between 0 and 1 which ended up improving the accuracy rate of our machine learning model. Because of data scaling, the machine learning model does not give more priority to a specific column due to its weight as all the values are now between 0 and 1. As a result, the learning process gets better and more efficient.

Models applied

Since our goal is to predict whether or not someone had lung cancer, it is a classification problem. We have used the following classifiers in our project to compare their performances.

1) Naive Bayes

Naive Bayes is a type of classifier that is based on conditional probability. Here, we assume that all the features are conditionally independent. This makes it easier to compute the result for larger datasets as we do not have to consider dependency for a large number of input features.

In the training phase, the conditional probability of all input features with regards to each output label is found out. Afterwards, this information is used to predict the outcome of different combinations of input features. The formula for Naive Bayes is P(C|X) = (P(X1|C) P(X2|C) P(X3|C)... P(Xn|C)) P(C)

For our dataset, let us consider a combination of inputs,

X= (Gender = 1, Smoking = 2, Anxiety = 2, Chronic Disease = 1… Chest Pain = 1) where 1 = no and 2 = yes.

In this case, the output label will be decided in the following way-

P(yes|X) = (P(1|yes) P(2|yes) P(2|yes) P(1|yes)... P(1|yes)) P(yes)

P(no|X) = (P(1|no) P(2|no) P(2|no) P(1|no)... P(1|no)) P(no)

Depending on which posterior probability is higher, the label will be set accordingly.

We have chosen the Gaussian Naive Bayes classifier for our project. It is good for predictions from normally distributed features.

2) Random Forest

Random forest is a kind of Ensemble Classifier which focuses on producing better predictions using many learning algorithms that perform better together than they could individually. This classifier consists of a large number of decision trees that operate together. A random forest can have many decision trees within it and it is set using the n_estimator parameter. In our project, we have left this value as default which is 100. Larger n_estimator gives more accurate results but it is slower.

Each of the uncorrelated decision trees produces a prediction of its own and the result is determined by taking the average of these predictions or by majority voting. Random forest does not follow any set of formulas. Features are randomly selected and decision trees are created. Because of this, every decision tree is different from the other. The trees being uncorrelated plays a vital role in achieving high accuracy because it limits errors within the trees. If one tree faces an error, it will not affect the accuracy and performance of the others. Random forest is useful for datasets with a large number of features because even though a lot of trees are being used, it doesn't overfit the model.

3) Logistic Regression

Logistic Regression is a classification algorithm which is used to calculate the probability of an event occurring . The nature of the event occurring is usually binary and can have two possible values which are 0 and 1. Here 0 indicates that an event is not occurring and 1 indicates that an event is occurring. For example in this project of predicting lung cancer using machine learning techniques we have used logistic regression to predict whether a person has lung cancer or not. If a person has lung cancer then the value of the column lung cancer is 1, otherwise the value of the lung cancer column is 0 in order to show that a person does not have lung cancer.

There are three types of logistic regression which are binary logistic regression, multinomial logistic regression and ordinal logistic regression. In our model we have used binary logistic regression as through this model it is being predicted whether a person has lung cancer or not and while applying logistic regression some assumptions have been made. The assumptions made while applying logistic regression are that the predicted result is binary, the dependent variable is categorical in nature, the independent variable does not have multicollinearity and sufficiently large sample sizes are used.

We have used logistic regression for our project as the data is discrete and using logistic regression we can predict the likelihood of a person being affected with lung cancer.

4) K - Nearest Neighbor (KNN)

K - Nearest Neighbor (KNN) is a nonparametric supervised learning classifier. This algorithm classifies a single point into a group based on the distance between the single point and the nearest neighbors of that point. In KNN the symbol "K" denotes the number of closest neighbors to a new point that needs to be predicted or categorized into a group. The KNN algorithm can be used to solve classification and regression problems. As our project is a classification problem which requires us to predict lung cancer, hence we have used KNN as a classification algorithm.

In order to implement KNN, at first we need to choose a value of K, where the value of K will be an integer number and it denotes the number of neighbors that we will take into consideration. After selecting the value of K, we will have to calculate the distance between the single point and the K number of points using different methods such as Euclidean distance, Manhattan distance, Hamming distance etc. After calculating the distance between the single point and neighboring points we need to select the K nearest neighbors according to the calculated distance using one of the distance metrics mentioned above. Then we need to count the number of data points among these K nearest neighbors and then we will categorize the single data point to a category where the number of nearest neighbors is maximum.The KNN model will be ready after we have grouped the new single data points into different categories.

For example in this project the data set can be divided into two categories where one category represents a person not having lung cancer and the other category represents a person having lung cancer. In this example let's consider that the value of K that we have chosen is K= 5. Now suppose we have a single point near these two categories and we need to classify the point into a category. At first we will use distance metrics such as the Euclidean distance, Manhattan distance, Hamming distance to calculate the distance between the single point and its neighboring points. After calculating the distance we will select 5 nearest neighbors of the single point. Suppose after selecting the nearest neighbors of the single point we can see that 3 of the nearest neighbors of the single point belong to the category where lung cancer is present and 2 of the nearest neighbors of the single point belong to the category where lung cancer is not present. Now in order to classify the single point into the group we will need to select the category where the nearest neighbor is maximum. Here from this example we can see that there are 3 neighbors

in the lung cancer present category and 2 neighbors in the lung cancer not present category, hence it can be seen that the maximum neighbors of the single point belong to the category of lung cancer. Therefore this single point belongs to the lung cancer present category and will be grouped as lung cancer present. For the rest of the new uncategorized data points we will follow the same procedure of KNN algorithm and group the points to a category according to its nearest neighbor.

For this project we have chosen to use KNN as a classifier because it uses a few hyperparameters such as the K value and the distance metric, the KNN algorithm can also modify easily when new data points are added and the algorithm is simple and easy to implement compared to other machine learning algorithms.

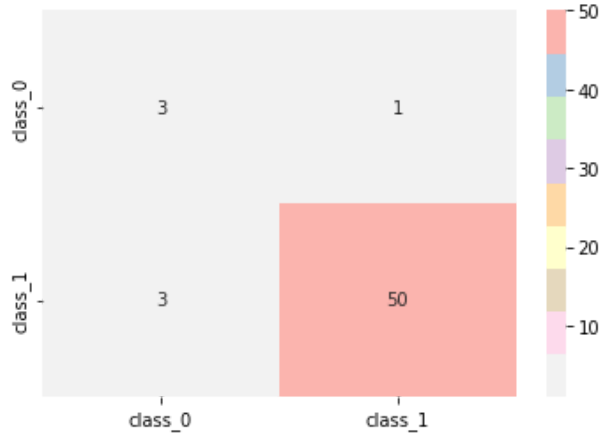
Model usage

For each of the models, first we create an object using the model class from SciKit Learn library and then use the fit method to fit our training and testing sets. Then we calculate the prediction using the predict method. Finally, we find the accuracy, F1, precision and recall scores using the metrics module and store them for comparison.

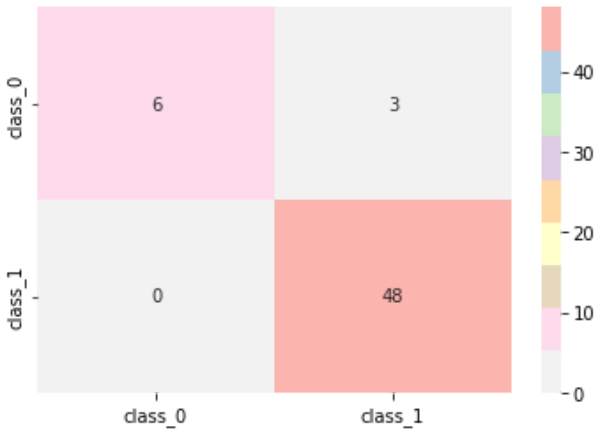
Results

Confusion matrix

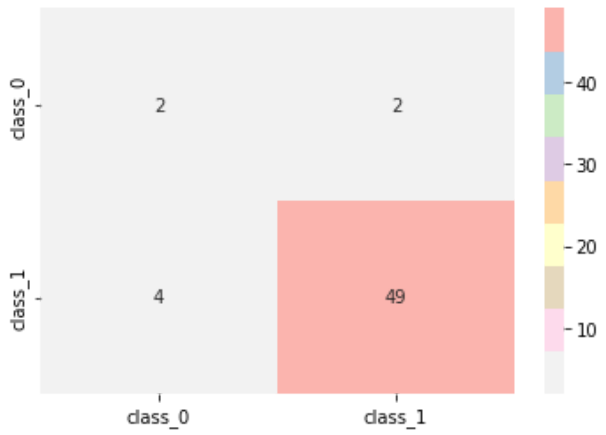
A confusion matrix is a table that is used to describe the performance of a classification model on a set of test data for which the true values are known.It makes it easy to see if the model is confusing classes.



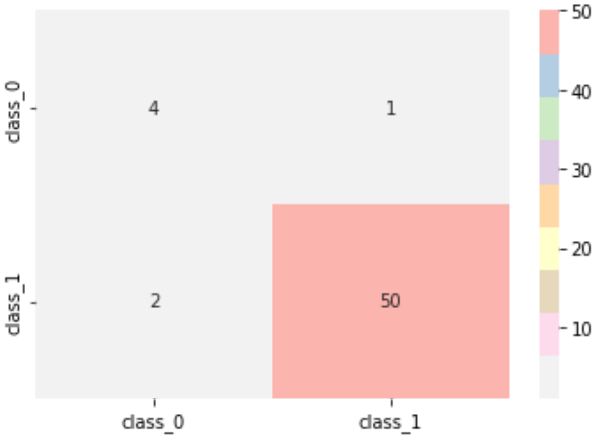
Logistic regression



KNN



Naive Bayes



Random Forest

Based on the confusion matrices of all our models, we can see that all models have performed well and none has confused two classes.

Accuracy score, F1 score, Precision score and Recall score

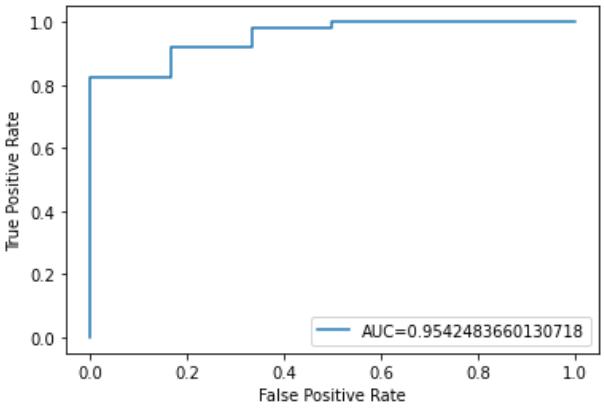
We can use some metrics like accuracy score, precision score, recall score and F1 score to measure the performance of our models. Accuracy score is the ratio of correctly classified data instances and the total number of data instances. Accuracy is not a good metric when it comes to an unbalanced dataset. Precision is the ability of a classifier to label a negative sample as negative. Recall is the ability of the model to predict the positives out of samples that are truly positives. F1 takes both precision and recall into account so if precision score and recall score are both high, F1 score will be high as well. For all metrics, 1 is the best result and 0 is the worst.

We have made a table with all the metrics that we had stored before. We can see the performances of the models at a glance from this table.

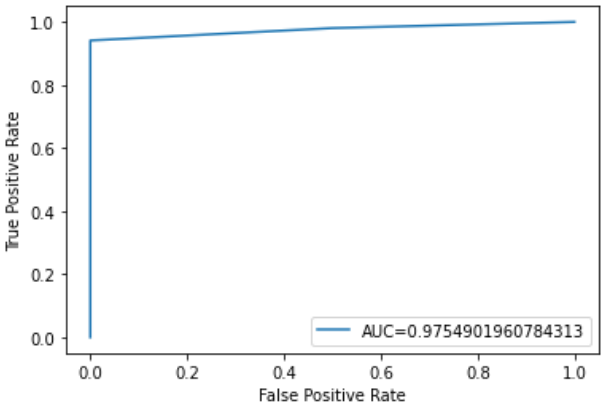
Model	Accuracy score	F1 score	Precision score	Recall score
Logistic Regression	0.93	0.96	0.94	0.98
KNN	0.95	0.97	1.00	0.94
Naive Bayes	0.89	0.94	0.92	0.96
Random Forest	0.95	0.97	0.96	0.98

ROC Curve

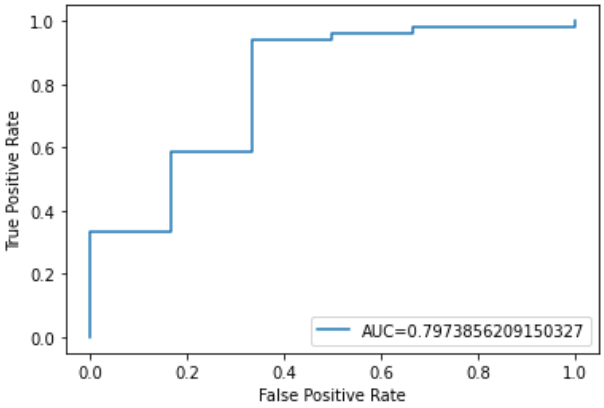
An ROC curve (receiver operating characteristic curve) is a graph showing the performance of a classification model at all classification thresholds. We can visualize the True Positive Rate (TPR) and False Positive Rate (FPR) of a model by using the ROC curve. An ROC curve plots TPR vs. FPR at different classification thresholds. AUC (Area Under the Curve) is the entire two dimensional area under the curve. The higher the value of AUC, the more accurate the result is.



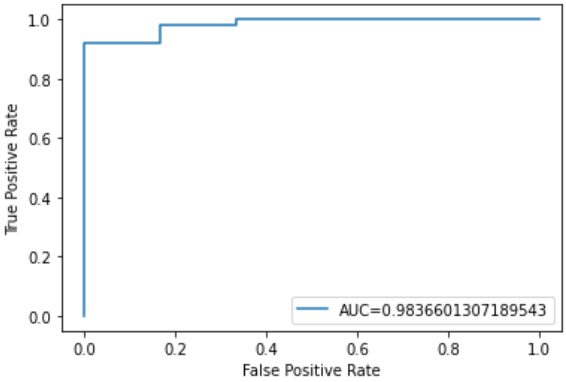
Logistic regression



KNN



Naive Bayes



Random Forest

From the table and the ROC curves, we can see that Logistic regression, KNN and Random Forest have produced very good results. Their scores vary slightly across the different metrics but overall accuracy is very satisfactory. KNN has a precision score of 1 which is excellent. Naive Bayes has performed well but it is not as accurate as the others.

References

https://www.statology.org/plot-roc-curve-python/

https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc/

https://www.geeksforgeeks.org/display-the-pandas-dataframe-in-table-style/

https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/

https://www.ibm.com/topics/knn#:~:text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.

https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning

https://www.capitalone.com/tech/machine-learning/what-is-logistic-regression/

Dataset link : https://www.kaggle.com/datasets/nancyalaswad90/lung-cancer