Thesis Book for Lung cancer prediction using machine learning algorithms

Abstract

This is the overall summary of our thesis book.

**Keywords:** Lung Cancer, Machine Learning.

Contents

Chapter 1: Introduction

* 1. Introduction…………………………………………………… 2
  2. Introduction…………………………………………………… 3
  3. Introduction…………………………………………………… 3
  4. Introduction…………………………………………………… 3
  5. Introduction…………………………………………………… 3

Chapter 2: Literature Review

* 1. Introduction…………………………………………………… 2
  2. Introduction…………………………………………………… 3
  3. Introduction…………………………………………………… 3
  4. Introduction…………………………………………………… 3
  5. Introduction…………………………………………………… 3

Chapter 3: System Architecture & Data Analysis

* + 1. Introduction…………………………………………………… 2
    2. Introduction…………………………………………………… 3
    3. Introduction…………………………………………………… 3
    4. Introduction…………………………………………………… 3
    5. Introduction…………………………………………………… 3

Chapter 4: Machine Learning Approach & Proposed Models

* 1. Introduction…………………………………………………… 2
  2. Introduction…………………………………………………… 3
  3. Introduction…………………………………………………… 3
  4. Introduction…………………………………………………… 3
  5. Introduction…………………………………………………… 3

Chapter 5: Result & Discussion

* 1. Introduction…………………………………………………… 2
  2. Introduction…………………………………………………… 3
  3. Introduction…………………………………………………… 3
  4. Introduction…………………………………………………… 3
  5. Introduction…………………………………………………… 3

Chapter 6: Conclusion

* 1. Introduction…………………………………………………… 2
  2. Introduction…………………………………………………… 3
  3. Introduction…………………………………………………… 3
  4. Introduction…………………………………………………… 3
  5. Introduction…………………………………………………… 3

References…………………………………………………… 65

Chapter 1: Introduction

Chapter 2: Literature Review

Chapter 3: System Architecture & Data Analysis

In this chapter of our book we are going to discuss from where we collect our dataset and how we analyze the dataset and all of its attributes.

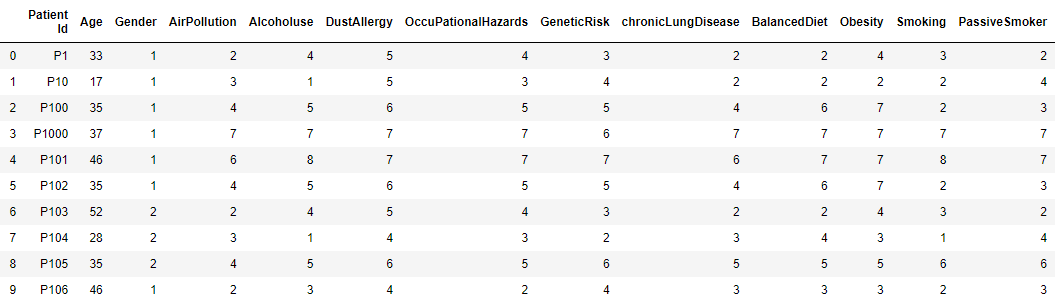
3.1 System Design

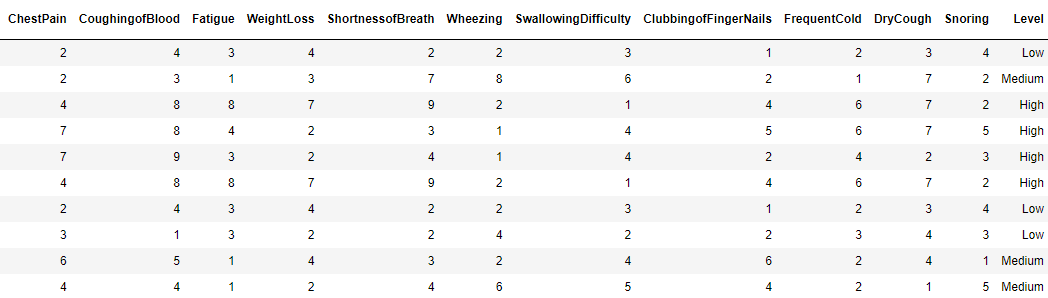
3.2 Dataset

The dataset which is used for our machine learning models is taken from Kaggle (<https://www.kaggle.com/bilalatiq/cancerpatientsdata>).

It contains 1000 records and 25 features or columns. Among 25 features 24 features are independent features and one is dependent feature. The dependent feature is the result or output column of our dataset. By 24 independent features our machine learning models will predict the lung cancer condition of a patient. Every records of our data set represent a patient. Our dataset is a comma separated values (csv) file.

3.2.1 Sample of Our Dataset:





3.3 Features Analysis:

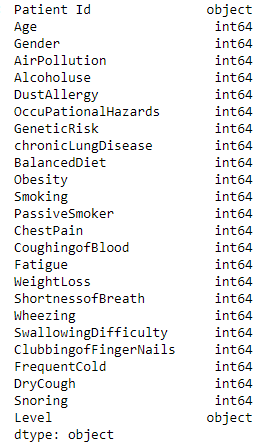
In this section we are going to discuss about the features of our data set. In general feature or column represents an observable piece of data that can be analyzed. Ex Age, Gender, Alcoholuse, ChestPain, and so on for our dataset.

Features are also known as "variables" or "attributes" in some contexts. The features include in our dataset contains the following values

1. Numerical value
2. String value

“Patient Id” and “Level” columns of our dataset have string value and others columns have numerical value. Patient id represent the unique identity for every patient. Level column represent the status of the result by storing Low, Medium and High value for every patient. Except these two feature others contain the numerical value which represent the patient health condition and based on the value of these features our models will predict the condition of lung cancer for particular patient.

3.4 Data type of each columns:



# 3.5 Data Preprocessing

Data Preprocessing is the initial activity for building machine learning models after collecting data. It is the process or technique of arranging raw data when used with a machine learning model. It is indeed the most important step towards developing a machine learning model.

# When we are working to develop a machine learning project, we do not always come across tidy and formatted data. And, before trying to perform any specific operation on data, it must be cleaned and formatted for the specific operation. As an outcome, we do the data pre-processing task and it is the most important and lengthy portion of working area while building machine learning model.

3.5.1 Import Dataset

To pre-process our dataset first of all we have to import dataset. For importing dataset we use pandas. Pandas is a well-known Python-based data analysis toolkit. It comprises a wide variety of utilities, from parsing numerous file formats to converting an entire data table to a numpy matrix array. As a consequence, pandas has earned a reputation as a reliable ally in data science and machine learning.

Pandas mainly works on data in 1-D and 2-D arrays. These two arrays are handled differently in pandas.

3.5.2 Check Duplicate or Null Value

If dataset contains any null or duplicate values then this will be a concern for the accuracy of machine learning model. So we check for null or duplicate value. Our dataset does not contain any null or duplicate value.

3.5.3 Cases in Dataset

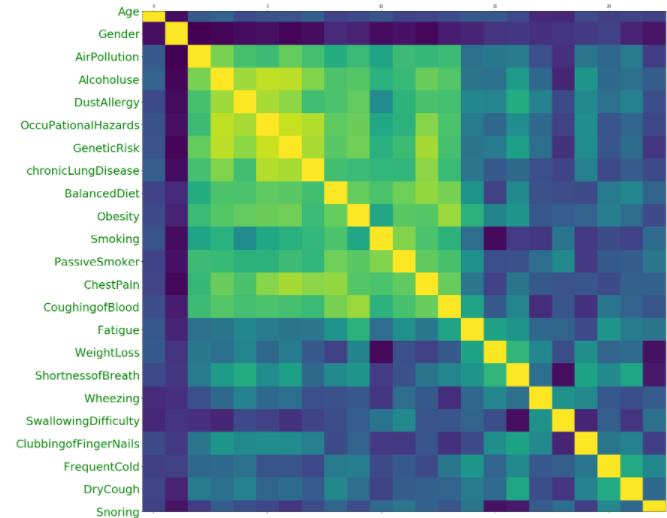
Our Dataset contains three type of cases, Low, Medium and High. For more control over dataset we have to know the dataset’s cases ratio.

|  |  |  |
| --- | --- | --- |
| Level | Total Cases | Ratio (%) |
| Low | 303 | 30.30 |
| Medium | 332 | 33.20 |
| High | 365 | 36.50 |

3.5.4 Correlation Heatmap

We can be define correlation heatmap as a visual representation of the correlation matrix. It is the illustration of the relationship between different variables of the corresponding dataset. The correlation value can range from -1 to 1. By correlation heatmap we can easily visualize the relation among the features of dataset and find out the correlated columns. If there are more correlation available then the correlated column will be drop except the one form them. There are no more correlated columns in our dataset. So we don’t drop any features.

3.5.4.1 Correlation Heatmap of our dataset



3.5.5 Modify and Drop some columns

In our working dataset we have a dependent column which contain the patient unique identity. This column does not have any impact on learning and accuracy of our model. So we drop this column.

We also modify the predict column which is called “level”. This column contains value in strings format. We replace the values of this columns by 0 for Low, 1 for Medium and 2 for High.

3.6 Features Selection

In practice, it is rare that all of the variables in a dataset are useful for building a machine learning model. The redundant variables decrease the model's prediction performance and may also decrease a classifier's accuracy rate. Adding more variables to train a model increases the model's overall complexity. In machine learning, feature selection technique is used to find out the more impactful features which has more score to impact on the prediction of model. By selecting feature, total number of features are decreased. So the complexity of the model training also decrease.

Before features selection we calculate the every columns score of our dataset by SelectKBest model of feature\_selection of sklearn library.

3.6.1 Score of all columns

Ekhane Screenshot dibo column er score gulor.

Form 23 dependent columns of our dataset we choose the best 10 columns which have more impact on the prediction of various machine learning model.

3.6.2 Selected Columns

Here are the list of selected columns which are more responsible to predict the lung cancer with the great accuracy.

Ekhane Screenshot dibo column er score gulor.

3.7 Dataset Splitting

After pre-processing the corresponding dataset, next important task is to split the dataset. In our thesis we split our dataset by the ratio of 70% of data for training and 30% of data for testing. Overfitting and underfitting are the two major factors that happen in machine learning and debase the effectiveness of the training models. To avoid the overfitting and underfitting we have need more data for training. Another way of overcome this we use different types of cross validation. In our thesis we use following three types of cross validation to avoid the overfitting and underfitting of the model.

1. Leave one out cross-validation.
2. K-Fold cross-validation.
3. Stratified k-fold cross-validation

In general we split our dataset for building machine learning model into two groups called train data and test data.

3.7.1 Train Data

Train data is the portion of our dataset which is used to train model. Model actually sees the both input and output of the train data and learn from the train data.

3.7.2 Test Data

Test Data is the portion of our dataset which is used to test the accuracy of model after training them by train data. Model don’t see the output of this data. Model learn from the train data and based on the learning form train data it’s predict from the test data. Outcomes from the test data we identify the accuracy of the particular model.

Chapter 4: Machine Learning Approach & Proposed Models

4.1 Machine Learning

Machine Learning is the study of how to teach computers to learn without explicitly programming them. Machine learning is one of the most trending technologies that I have ever encountered. As the name implies, it provides the computer with a feature that makes it more human-like: the capacity to learn. Machine learning is being used actively today, possibly in many more areas than one would anticipate. Machine learning is a branch of artificial intelligence that is broadly defined as a machine's ability to mimic intelligent human behavior. Ai systems are being used to solve complex problems in a manner similar to how people deal with problems.

“Machine learning is the study of computer algorithms that allow computer programs to automatically improve through experience”

~Tom Mitchell

4.2 Machine Learning Algorithms

There are four types of machine learning algorithms are available.

1. Supervised
2. Semi-supervised,
3. Unsupervised and
4. Reinforcement.

4.2.1 Supervised learning

The machine is trained by example in supervised learning. The operator gives the machine learning algorithm a known dataset with intended inputs and outputs, and the method must figure out how to get those inputs and outputs. While the operator is aware of the correct remedies, the algorithm recognizes data patterns, learns from observations, and makes predictions. The algorithm makes predictions, which are then corrected by the operator, and this process is repeated until the classifier accomplishes a significant amount of accuracy.

Supervised learning can be classified into three category.

4.2.1.2 Classification

In classification supervised learning algorithms machine learning program must deduce an outcome from observations and determine which category new observations belong to. For example, predicting that patients have lung cancer or not. The program must look at existing observational data and filter the data accordingly. Classification algorithms are used when the output variable is categorical, discrete value like 1-100, 200-300 or small, medium, large.

4.2.1.2 Regression

When the input and the output features have a correlation, regression algorithms are used. It is used to forecast dependent variable such as weather, market price, and so on. The machine learning program should always estimate and comprehend the relationships among the variables in regression tasks. Regression analysis is very useful for modeling and analysis because it tends to focus on one predictor variables and a series of many other changing variables.

4.2.1.3 Forecasting

Forecasting is a method of making future predictions based on available data, and it is commonly often used discover patterns.

4.2.2 Semi-supervised

Semi-supervised learning is equivalent to supervised learning that it employs both labeled and unlabeled data. Labeled data is the data that has relevant tags so that the algorithm can acknowledge it, whereas unlabeled data does not have that information. Algorithms can gain knowledge to label and unlabeled data using this fusion. Semi-supervised machine learning is a perfect fusion of supervised and unsupervised learning methodologies. It employs a tiny amount of labeled data and a large quantity of unlabeled data, allowing it to take advantage including both unsupervised and supervised learning algorithms while attempting to avoid the difficulties associated with locating a large amount of labeled data. As a result, you can train the model to label data without ever using quite so much labeled training data.

4.2.3 Unsupervised learning

In this case, the machine learning algorithm examines data in order to identify patterns. There is no response key or human operator to provide guidance. Instead, by analyzing available data, the machine determines correlations and relationships. In an unsupervised learning, the machine learning algorithm is left to perceive huge amounts of data and address that data accordingly. The algorithm attempts to organize that data in order to describe its structure. This could imply categorizing the data into clusters or organizing it in a more effective fashion. Unsupervised learning, as the name implies, is a machine learning technique in which models are not supervised using training datasets. Instead, models reveal hidden patterns and insights in the corresponding data.

4.2.4 Reinforcement

This is worried with routinized learning processes where a machine learning algorithm is offered a set of actions, parameters, and end values to implement. Following the definition of the rules, the machine learning algorithm attempts to explore different options and possible scenarios, implementing and assessing each result to determine which is optimal. Reinforcement learning instructs the machine through trial and error. It learns from prior experience and commences to make adjustments its strategy in reaction to the situation in order to obtain the best possible outcome.

4.3 Algorithm Selection and modelling

The main goal of the entire thesis is to predict lung cancer condition with the highest accuracy. Our problem is a classification problem cause outcomes of this will be the condition of lung cancer in Low, Medium or High. In this thesis we are going to use several classification algorithms. We have chosen several algorithms typical for solving supervised learning problems throughout classification methods. In this stage we have preprocessed dataset and decided to use supervised classification algorithms.

Next step is to build our machine learning model. To build machine learning model we use sklearn library, from this library we import our selected algorithms and train them with our train data using fit function.

The various classification algorithms will be covered in the following section.

4.3.1 Support vector machine

Support vector machines are the type of supervised learning algorithm which is most used for classification, regression, and outlier detection. All those are conventional learning tasks. We can use them to detect cancerous cells using thousands of images, or we can use a well-fitted regression model to forecast future cruising routes. SVMs of different kinds can be used to solve conventional machine learning problems, including support vector regression (SVR), which is a support vector classification augmentation (SVC). The important thing to remember here is that these are simply math equations that have been tuned to give you the most precise response possible as quickly as possible.

It diverge from the other classification algorithms in that way they can select the decision boundary that maximizes the distance from the nearby data points among all classes. The maximum margin hyper plane is the decision boundary engendered by SVMs.

A simplistic linear SVM classifier connects two classes by drawing a line between them. That is, every one of the data points on one side of the line will resemble a category, while the data points on the other side of the line will be assigned to a separate category. This means that there could be an infinite number of lines to choose from. The difference of linear SVM algorithm and others including such k-nearest neighbor is that it chooses the best line to categorise the data points. It chooses the way to distinguish the data and is as much further away from the nearest data point as possible.

Types of SVMs:

There are two types of SVMs, each used for different purpose:

**Simple SVM:** This type of SVM is used only for linear classification and regression analysis.

**Kernel SVM:** This type of SVM is used for non-linear data, because it can fit a hyperplane instead of a two-dimensional space.

4.3.2 Logistic regression

This is a widely used learning algorithm under the category of supervised learning. It is used to estimate the categorical dependent feature from a set of independent features. A categorical dependent variable's output is predicted using logistic regression. As a consequence, the result must be a categorical or piecewise value. It can be Yes or No, 0 or 1. Instead of providing the accurate value it delivers the probabilistic values. Logistic Regression and Linear Regression seem to be very equivalent. Logistic regression can be used to alleviate classification types problems on the other hand linear regression algorithm can be used to remedy regression type problems.

Mechanism of logistic regression is that it does not fit a regression line, it fits a 'S' shaped logistic function that predicts two maximum values in logistic regression: 0 and 1. It is an essential machine learning algorithm which provide probabilities and characterize new data from both continuous and discrete datasets. Logistic Regression also be used to classify observations that used a variety of data types and can quickly select the most effective features for classification. Logistic Regression estimates likelihoods using its underpinning logistic function to ascertain the correlation between the variables, the output, and the independent variables. It utilises the L2 penalty for normalization. The logistic function, also renowned as the sigmoid function, generates the resulting probabilities to binary values 0 or 1. The sigmoid function converts any real-valued number into a value between 0 and 1, with the exception of the limits themselves. Following that, a threshold classifier converts the result to a binary value.  One of the core assumptions of Logistic Regression is that the input features be independent of one another.

4.3.3 Decision Tree Classifier

This is a member of the supervised learning algorithm family. Unlike other supervised learning algorithms, it can also be used to rectify regression and classification tasks. The main objective of using this algorithm is to construct a training model that predicts value of a target feature by gaining knowledge from simple decision rules from training data. In this algorithm we start from the root of the tree to predict target value for a record or row. We make comparison of the root attribute and the attributes of the record or row. It obey the branch that pertains to that value and step to the next node analysis and the comparison.

The factor in choosing strategic splits has a substantial impact on a tree's accuracy. The decision criteria for classification and regression trees are clearly different.

To make a decision whether it should split a node into two or more sub-nodes, decision trees implement various algorithms. The emergence of sub-nodes enhances the uniform distribution of the sub-nodes that result. In other words, the integrity of the node enhances in relation to the target variable. The decision tree segregates the nodes based on all available variables and then can choose the split that yields the most uniform sub-nodes. The type of dependent variable also effects algorithm selection. Here is a list of some algorithms used throughout Decision Trees.

### **Types of Decision Trees:**

Classification of this supervised learning algorithm is depend on the type of the target feature. There are two types of decision tree

1. Categorical Variable Decision Tree: A decision tree with such a categorical target variable is regarded to as a Categorical variable decision tree.
2. Continuous Variable Decision Tree:A decision tree with such a continuous target variable is regarded to as a Continuous variable decision tree.

4.3.4 k-nearest Neighbors

The k nearest neighbor algorithm is a Supervised Learning algorithm that is most commonly used for classification and regression. It is a powerful and flexible algorithm that can also be used to input missing values and resample datasets. As the name implies, k-nn takes into account K Nearest Data Points to predict the class or continuous value for the new data point.

**Learning Types of k-nn:**

1. Instance-based learning: Rather than learning weights from training data to predict output (as in model-based algorithms), we use entire training instances to predict output for previously unseen data.
2. Lazy Learning: The model is not learned using training data before the prediction is requested on the new instance, and the learning process is postponed until the prediction is requested.
3. Non-Parametric: There is no predetermined pattern of the mapping function in knn.

Consider the following figure. We have plotted two types sample data points from on a two-dimensional feature space. As shown, we have a total of 9 data points 5 blue circle and 4 yellow circle. Blue data points belong to ‘class A’ and yellow data points belong to ‘class B’. And yellow star data point in a space represents the new point for which a class is to be predicted. Obviously, we say it belongs to ‘class B’ yellow points.

Class A Class B

This is the underlying principle of K Nearest Neighbors. In this case, nearest neighbors are those data points that are the closest in feature space to our new data point. And K is the number of such data points that we consider in our algorithm implementation. As a consequence, the distance metric and K value are two critical considerations while employing the KNN algorithm. The most broadly used distance metric is the Euclidean distance. Depending on the requirements, you may use Hamming distance, Manhattan distance, and Minkowski distance.

4.3.4 Naïve Bayes

It is indeed a classification algorithm based on Bayes' Theorem and the assumption of estimator independence. A Naive Bayes classifier, in simple terms, implies that the presence of one feature in a class is independent of the presence of any other characteristic. The Naive Bayes model is simple to implement and is incredibly beneficial for very large amounts of data. In additament to its easiness, Naive Bayes has been shown to outcompete even the most sophisticated classification methods.

Bayes Theorem:

P(A|B)

**Where,**

**P(A|B) is Posterior probability**.

**P(B|A) is Likelihood probability**.

**P(A) is Prior Probability**.

**P(B) is Marginal Probability.**

### **Types of Naive Bayes:**

**1. Gaussian Naïve Bayes:**While characteristic values are continuous, it is hypothesised that the values associated with each class are evenly distributed according to the Gaussian distribution, also known as the Normal Distribution.

**2. Multinomial Naïve Bayes:**On multinomial distributed data, Multinomial Naive Bayes is preferred. It is widely used in NLP text classification. Each text classification event represents the presence of a word in a document.

**3. Bernoulli Naïve Bayes:**Bernoulli Naive Bayes can be used when data is spread according to multivariate Bernoulli distributions. That is, multiple features exist, but each is assumed to have a binary value.

4.3.4 Random Forest Classifier

Chapter 5: Result & Discussion

5.1 Introduction

Many more tasks have been done yet, include data pre-processing, data splitting, model selection, model train etc. Now it’s time to discuss about the result of our proposed models. In this chapter of our thesis book we are going to discuss about the result of each chosen model and show the comparison of them.

5.2 Performance

Performance is the accuracy of a model which indicate that a particular model how much capable to predict in the corresponding domain with great accuracy. There are many way to measure the performance of machine learning models. In the following sections we are going to discuss performances metrics to evaluate the performance of our chosen model.

5.3 Performance Metrics

Among different metrics, performance metrics are used to evaluate the performance of machine learning classification algorithms. Since our problem is a classification type problems and our selected models are classification type so we are going to use different performance metrics to evaluate the performance of our models.

5.3.1 Confusion Matrix

A Confusion matrix is a N x N matrix that is being used to examine the accuracy of the model, where N is the number of target classes. The matrix compares the actual target values to the machine learning model's predictions. This gives us with a complete view about how well our classification model is performing and the types of issues it is making.

|  |  |  |  |
| --- | --- | --- | --- |
| Confusion Matrix | | Predicted | |
| Negative | Positive |
| Actual | Negative | TN | FP |
| Positive | FN | TP |

**1. True Positive:**

The predicted value matches the actual value.

The actual value was positive and the model predicted a positive value.

**2. True Negative:**

The predicted value matches the actual value.

The actual value was negative and the model predicted a negative value.

**3. False Positive:**

The predicted value was falsely predicted.

The actual value was negative but the model predicted a positive value.

**4. False Positive:**

The predicted value was falsely predicted.

The actual value was positive but the model predicted a negative value.

**5.3.3 Accuracy:**

It is a performance metric for classification type algorithms. It may define the total number of correct prediction over the total number of predictions.

Accuracy =

**5.3.3 Precision:**

It is a performance metric for classification type algorithms. It may define how many of the correctly predicted cases were actually positive.

Precision =

**5.3.4 Recall/ Sensitivity:**

It is a performance metric for classification type algorithms. It may define how many of the actual positive cases we were able to predict correctly with our model.

Recall/ Sensitivity =

**5.3.5 Specificity:**

It is a performance metric for classification type algorithms. It is the opposite of Recall/ Sensitivity. It may define the number of negative cases we were able to predict correctly with our model.

Specificity =

**5.3.6 F1 Score:**

It is a performance metric for classification type algorithms. It is a **harmonic mean of Precision and Recall.**

F1-Score =

**5.3.7 ROC Curve:**

It is a performance metric for classification type algorithms. It is created by plotting sensitivity/recall vs 1-specificity at different threshold values. The ROC is depicted in the figure below. On the y-axis, recall/sensitivity is plotted, and on the x-axis, 1- Specificity is plotted.

5.4 Model Performance

In this section of our thesis book we are going to discuss the performance of individual model. Ekhane aro likhbo

**5.4.1 Support vector machine:** ekhane kichu likhbo.

**Performance Metrics:**

|  |  |  |
| --- | --- | --- |
| Performance Metrics | With All Features  (%) | With Selected Features (%) |
| Accuracy | 99.667 | 100.00 |
| Precession | 100.00 | 100.00 |
| Recall | 100.00 | 100.00 |
| F1-Score | 100.00 | 100.00 |
| AUC | 100.00 | 100.00 |

**Cross Validation:**

|  |  |  |
| --- | --- | --- |
| No | Name of Cross Validation | Accuracy (%) |
| 01 | K-Fold Cross | 99.90 |
| 02 | Stratified Cross | 99.90 |
| 03 | Leave One Out Cross | 99.90 |

**Confusion Matrix:**

**Screen shoot hbe ekhane (with and without features selection)**

**ROC Curve:**

**Screenshoot hbe ekhane (with and without features selection)**

**5.4.2 Logistic regression:** **ekhane kichu likhbo.**

**Performance Metrics:**

|  |  |  |
| --- | --- | --- |
| Performance Metrics | With All Features  (%) | With Selected Features (%) |
| Accuracy | 97.333 | 96.33 |
| Precession | 97.00 | 97.00 |
| Recall | 97.00 | 96.00 |
| F1-Score | 97.00 | 96.00 |
| AUC | 97.70 | 97.20 |

**Confusion Matrix:**

**Cross Validation:**

|  |  |  |
| --- | --- | --- |
| No | Name of Cross Validation | Accuracy (%) |
| 01 | K-Fold Cross | 98.50 |
| 02 | Stratified Cross | 98.50 |
| 03 | Leave One Out Cross | 98.40 |

**Screen shoot hbe ekhane (with and without features selection)**

**ROC Curve:**

**Screenshoot hbe ekhane (with and without features selection)**

**5.4.3 Decision Tree Classifier:** **ekhane kichu likhbo.**

**Performance Metrics:**

|  |  |  |
| --- | --- | --- |
| Performance Metrics | With All Features  (%) | With Selected Features (%) |
| Accuracy | 100.00 | 100 |
| Precession | 100.00 | 100.00 |
| Recall | 100.00 | 100.00 |
| Specificity | 100.00 | 100.00 |
| F1-Score | 100.00 | 100.00 |
| AUC | 100.00 | 100.00 |

**Confusion Matrix:**

**Cross Validation:**

|  |  |  |
| --- | --- | --- |
| No | Name of Cross Validation | Accuracy (%) |
| 01 | K-Fold Cross | 100.00 |
| 02 | Stratified Cross | 100.00 |
| 03 | Leave One Out Cross | 100.00 |

**Screen shoot hbe ekhane (with and without features selection)**

**ROC Curve: Screenshoot hbe ekhane (with and without features selection)**

**5.4.4 k-nearest Neighbors: ekhane kichu likhbo.**

**Performance Metrics:**

|  |  |  |
| --- | --- | --- |
| Performance Metrics | With All Features  (%) | With Selected Features (%) |
| Accuracy | 99.667 |  |
| Precession | 100.00 |  |
| Recall | 100.00 |  |
| F1-Score | 100.00 |  |
| AUC | 99.70 |  |

**Confusion Matrix:**

**Cross Validation:**

|  |  |  |
| --- | --- | --- |
| No | Name of Cross Validation | Accuracy (%) |
| 01 | K-Fold Cross | 99.80 |
| 02 | Stratified Cross | 99.80 |
| 03 | Leave One Out Cross | 99.80 |

**Screen shoot hbe ekhane (with and without features selection)**

**ROC Curve:**

**Screenshoot hbe ekhane (with and without features selection)**

**5.4.4 Naïve Bayes: ekhane kichu likhbo.**

**Performance Metrics:**

|  |  |  |
| --- | --- | --- |
| Performance Metrics | With All Features  (%) | With Selected Features (%) |
| Accuracy |  |  |
| Precession |  |  |
| Recall |  |  |
| Specificity |  |  |
| F1-Score |  |  |
|  |  |  |

**Confusion Matrix:**

**Cross Validation:**

|  |  |  |
| --- | --- | --- |
| No | Name of Cross Validation | Accuracy (%) |
| 01 | K-Fold Cross | 98.50 |
| 02 | Stratified Cross | 98.50 |
| 03 | Leave One Out Cross | 98.40 |

**Screen shoot hbe ekhane (with and without features selection)**

**ROC Curve:**

**Screenshoot hbe ekhane (with and without features selection)**

**5.4.4 Random Forest Classifier: ekhane kichu likhbo.**

**Performance Metrics:**

|  |  |  |
| --- | --- | --- |
| Performance Metrics | With All Features  (%) | With Selected Features (%) |
| Accuracy |  |  |
| Precession |  |  |
| Recall |  |  |
| Specificity |  |  |
| F1-Score |  |  |
|  |  |  |

**Confusion Matrix:**

**Cross Validation:**

|  |  |  |
| --- | --- | --- |
| No | Name of Cross Validation | Accuracy (%) |
| 01 | K-Fold Cross | 98.50 |
| 02 | Stratified Cross | 98.50 |
| 03 | Leave One Out Cross | 98.40 |

**Screen shoot hbe ekhane (with and without features selection)**

**ROC Curve:**

**Screenshoot hbe ekhane (with and without features selection)**

**5.5 Summary of Result: ekhane kichu likhbo.­**

**Performance Metrics:**

|  |  |  |
| --- | --- | --- |
| Performance Metrics | With All Features  (%) | With Selected Features (%) |
| Accuracy |  |  |
| Precession |  |  |
| Recall |  |  |
| Specificity |  |  |
| F1-Score |  |  |
|  |  |  |

**Confusion Matrix:**

**Cross Validation:**

|  |  |  |
| --- | --- | --- |
| No | Name of Cross Validation | Accuracy (%) |
| 01 | K-Fold Cross | 98.50 |
| 02 | Stratified Cross | 98.50 |
| 03 | Leave One Out Cross | 98.40 |

**Screen shoot hbe ekhane (with and without features selection)**

**ROC Curve:**

**Screenshoot hbe ekhane (with and without features selection)**

5.6 Discussion:

Chapter 6: Conclusion

6.1 Future Scope

6.2 Conclusion

References