BREAST CANCER PREDICTION USING MACHINE LEARNING

A Project Report

Submitted in the partial fulfillment of the requirements

for the award of the degree of

Bachelor of Technology in

Department of Computer Science and Engineering

by

NAME	ID	SPECIALIZATION
MURIKIPUDI NIKHIL	190031117	DS&BDA (Cluster 7)

PROJECT BATCH - 171

UNDER THE ESTEEMED GUIDANCE OF

Dr.P. LAKSHMI PRASANNA

Associate Professor

Department of Computer Science and Engineering
K L Deemed to be University, Green Fields, Vaddeswaram,

Guntur District, A.P., 522 502.

K L University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

The Project Paper Report entitled **"BREAST CANCER PREDICTION USING MACHINE LEARNING"** is a record of bonafide work of 190031117 MURIKIPUDI NIKHIL submitted in partial fulfilment for the award of B. Tech in Computer Science and Engineering in K L University. The results embodied in this report have not been copied from any other departments/University/Institute.

190031117 MURIKIPUDI NIKHIL

K L University

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Project Report entitled **"BREAST CANCER PREDICTION USING MACHINE LEARNING"** is being submitted by 190031117 MURIKIPUDI NIKHIL, submitted in partial fulfilment for the award of B. Tech in Computer Science and Engineering to K L University is a record of bonafide work carried out under efficient guidance and supervision.

The results embodied in this report have not been copied from any other departments/University/Institute.

Signature of the Supervisor
Dr.P. LAKSHMI PRASANNA

Signature of the HOD

Signature of the External Examiner

Acknowledgment

We would like to express our deep gratitude to our project guide Dr. P. LAKSHMI PRASANNA, Associate Professor, Department of Computer Science and Engineering, KLEF, for her guidance with unsurpassed knowledge and immense encouragement. We are grateful to Dr Senthil Athithan, Head of the Department, Computer Science and Engineering, for providing us with the required facilities for the completion of the project work.

We are very much thankful to the Principal and Management, KLEF. for their encouragement and cooperation to carry out this work. We express our thanks to our professor in charge Dr G Siva Nageswara Rao, for his continuous support and encouragement.

We thank all teaching faculty of Department of CSE, whose suggestions during reviews helped us in accomplishment of our project. We would like to thank Mr. P. Nagesh of the Department of CSE, KLEF for providing great assistance in accomplishment of our project.

190031117 MURIKIPUDI NIKHIL

INDEX

ABSTRACT

In the developing world, cancer death is one of the major problems for humankind. Even though there are many ways to prevent it before happening, some cancer types still do not have any treatment. One of the most common cancer types is breast cancer. Early detection of breast cancer can dramatically improve the prognosis and chances of survival by allowing patients to receive timely clinical therapy. In this project, there are many studies about predicting the type of breast tumors. The detection of cancer cells is done by machine learning techniques. Machine learning techniques can bring a large contribute on the process of prediction and early diagnosis of breast cancer, became a research hotspot and has been proved as a strong technique.

All experiments are executed within a simulation environment and conducted in jupyter platform. The proposed work can be used to predict the outcome of different technique and suitable technique can be used depending upon requirement. This research is carried out to predict the accuracy.

INTRODUCTION

Breast cancer is a type of cancer that starts in the breast. Cancer starts when cells begin to grow out of control. Breast cancer cells usually form a tumor that can often be seen on an x-ray or felt as a lump. Breast cancer can spread when the cancer cells get into the blood or lymph system and are carried to other parts of the body. The cause of Breast Cancer includes changes and mutations in DNA. There are many different types of breast cancer and common ones include ductal carcinoma in situ (DCIS)[1] and invasive carcinoma. Others, like phyllodes tumors and angiosarcoma are less common. There are many algorithms for classification of breast cancer outcomes[5].

Machine learning plays an important role for the classification of the breast cancer. There are many diagnosis processes that have discussed above provides the images. These types of diagnosis images are used for the classification using machine learning. Machine learning is a sub-field of AI. Many developers use the machine learning to re-train the existing models and for the better performance. Machine learning is used for the linear data. If the data is small, then machine learning gives better results but when the data is too large then it doesn't give the better results. There are three main types of machine learning that are used to train the model. Supervised machine learning works on the known data and with the help of the supervisor. Unsupervised machine learning is taken without any supervision. Reinforcement machine learning is less in use. These algorithms catch the ideal information from past understanding to settle on the exact choices.

Using the concept of Deep Learning we can detect text from images using techniques such as Sliding window technique, Single Shot and Region based detectors, EAST (Efficient accurate scene text detector) and techniques such as CRNN[2], Machine Learning OCR with Tesseract which are used for recognition of text.

Only when magnitude of knowledge is very little, machine learning typically exhibits handy insights; conversely, when the portion of info is humongous, machine learning is dysfunctional. Any of three predominant types of machine learning is said to supervise the design. In attempt to craft benefits, ml with watchful eye uses before existing statistics and every aid of the advisor. For meticulous judgement call, these schemes extricate utmost vital data from old expertise. AI technology is used in ML, which permits the systems to automatically adapt from their experiences and get better eventually without having to be manually programmed. Building algorithms that can take input data and apply numerical analysis to determine an output while modifying outcomes as new data arrives is the fundamental idea behind ML. Another very critical activity of learning is referencing evident data or acquiring data, such as when seeking patterns in statistics to obtain forthcoming judgments based on.

The immediate aim is to empower technology train purely on their own, without encouragement or meddling by inhabitants, and change state accordingly. The axiom of DL provides a way to understand text from images using algorithms such as the sliding window approach, single shot and region-based detectors, EAST (Efficient Accurate Scene Text Detector), and techniques for text recognition such as CRNN and Machine Learning OCR using Tesseract. Our experiment's mission is to more accurately classify the individuals' cancer types into normal and cancerous forms via classification models. The Kaggle website provided the information. A criterion for machine learning which we've incorporated is supervised learning, in which creating the system independent and dependent categories to learn from, and then when the learning activity has been completed, the system predicts the significance of the dependent for a specified input in independent variable format. The prediction models used to find the carcinoma include DTree, K-NN, Support VM, and NBayes. We tested above models in a Jupyter notebook and also included data visualisation while analysing data.

LITERATURE SURVEY

This section gives the information about the related work of the research that has been already done. Basically, two techniques are used to detect the breast cancer. First one is machine learning and second is deep learning. There are many researches that are conducted through the machine learning[3]. But machine learning techniques have some problems that are removed through the deep learning. This section gives the information about machine and deep learning techniques.

Wang et al. [4] studied to find the best way for breast cancer predictions by using data mining methods on several records. They applied support vector machine (SVM), artificial neural network (ANN), naïve Bayes classifier, and AdaBoost Tree. Reducing the feature space was discussed, then Principle Component Analysis (PCA) was applied with the aim of reduction. In the evaluation part of the performance of the models, they used two datasets that were the Wisconsin Breast Cancer Database (1991) and Wisconsin Diagnostic Breast Cancer (1995) [4]. They provided a detailed evaluation of the models and test errors.

Nithya et al. [17]believe that the main problem of breast cancer is about classifying the breast tumor. Computer-Aided Diagnosis (CAD) has been used for the detection and characterization of breast cancer. Their main idea was improving breast cancer prediction by using data mining methods. Bagging, multiboot, random subspace to the classification performance of naïve Bayes, support vector machine-sequential minimal optimization (SVM-SMO), and multilayer perceptron were applied.

M. Tahmooresi et al. [7] proposed another hybrid model based on machine learning. According to that, SVM was a good classifier that gave the better accuracy among the all. It had done the comparison between SVM, KNN, ANN and decision tree. It implemented on the images and blood dataset[7]. As the consequence.

Muhammet Fatih Aslan et al. [6] proposed the model on machine learning but used the different classifier. The classifier used by the author was Extreme Learning Machine, SVM, KNN and ANN. There was a bit change in classifier to find out the better results. According to that, Extreme Learning Machine gave the better results[6].

Anusha Bharat et al. [10] proposed the model based on machine learning. It used the four classifier like SVM, decision tree (CART), KNN, Naïve bayes. According to author, KNN gave the better accuracy. There was limitation in SVM. SVM gave the better results for binary variables. So that's why Multi-SVM was used.

Ebru Ayndindag Bayrak et al. [9] done the comparison on machine learning techniques. The comparison was implemented on the WEKA and dataset was Wisconsin breast cancer dataset. According to author, SVM showed the better results in performance matrices. After the machine learning, deep learning techniques were developed to solve the problem of machine learning[9].

Shewtha K et al. [12] proposed the model on deep learning based convolution neural network. There were many models that came under the CNN but the used Mobile Net and Inception V3. Author had done the comparison on both models and found Inception V3 gave the better Accuracy. But there was a still chance to use machine learning for breast cancer.

Ch. Shravya et al. [18] proposed the model on supervised machine learning. This research was implemented on classifier like Logistic Regression, SVM and KNN. The dataset was downloaded from UCI repository and results were conducted with respect to performance. According to this, SVM was a good classifier that gave 92.7% accuracy on python platform.

LITERATURE TABLE

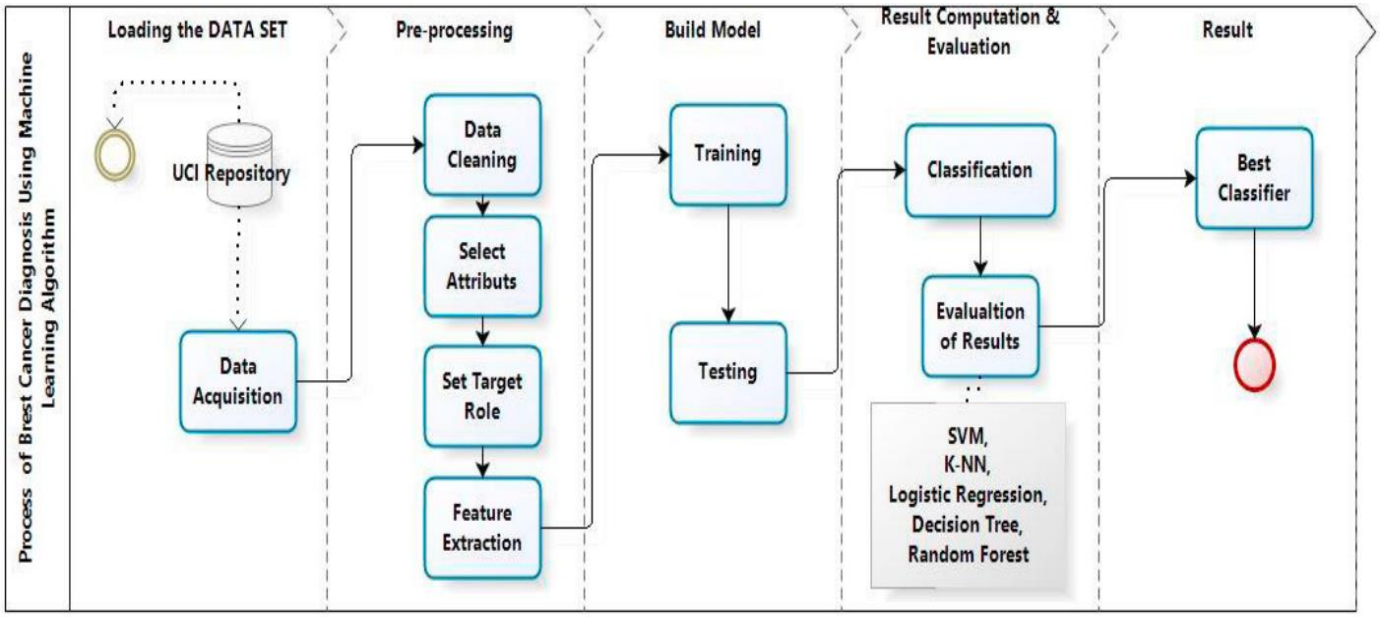
Author(s)	Year	Source	Title	Findings
Xiaojun Chen, Yixiang Fang, Min Yang, Feiping Nie	2018	IEEE Transactions on knowledge and data engineering	A Clustering Algorithm for Customer Segmentation from Massive Customer Transaction Data	In this paper, the authors have built a purchase tree for each customer and calculated novel separate density, and then select the top k customers as the representatives of k customer groups. Finally, the clustering results can be obtained by assigning each customer to the nearest representative tree.
Hongfu Liu, Jun Li, Yue Wu, Yun Fu	2019	IEEE Transactions on knowledge and data engineering	Clustering with Outlier Removal	In this paper, the authors considered the joint clustering and outlier detection problem and proposed the algorithm COR. Different from the existing K-means, first transformed the original feature space into the partition space according to the relationship between outliers and clusters. Then we provided the objective function based on the Holoentropy, which was partially solved by K-means optimization.
Dingming Wu, Jieming Shi, Nikos	2018	IEEE Transactions on knowledge and data engineering	Density-based Place Clustering Using Geo-Social Network Data	In this paper, the authors studied for the first time the problem of Density based Clustering Places in Geo-Social Networks (DCPGS). This clustering model extends the density-based clustering paradigm to consider both the spatial and social distances between places. We defined a new measure for the social distance between places, considering the social ties between users that visit them. Our measure is shown to be more effective to compute, compared to more complex ones based on node-to-node graph proximity and SNN-based model.
Ning Pang, Jifu Zhang, Chaowei Zhang, Xiao Qin	2019	IEEE Transactions on computers	Parallel Hierarchical Subspace Clustering of Categorical Data	In this paper, the authors investigated the problem of hierarchical clustering for categorical data. Their approach addresses the two problems encountered in the existing hierarchical clustering techniques. They proposed a Map Reduce-based hierarchical subspace clustering algorithm PAPU coupled with attribute-value weights using a data-partitioning strategy.

Ning Pang, Jifu Zhang, Chaowei Zhang, Xiao Qin	2019	IEEE Transactions on computers	Parallel Hierarchical Subspace Clustering of Categorical Data	In this paper, the authors investigated the problem of hierarchical clustering for categorical data. Their approach addresses the two problems encountered in the existing hierarchical clustering techniques. They proposed a Map Reduce-based hierarchical subspace clustering algorithm PAPU coupled with attribute-value weights using a data-partitioning strategy.
Xiaotong Zhang, Xianchao Zhang, Han Liu, Xinyue Liu	2018	IEEE Transactions on knowledge and data engineering	Partially Related Multi-Task Clustering	In this paper, the authors have proposed two multi-task clustering methods for partially related tasks: the self-adapted multitask clustering (SAMTC) method and the manifold regularized coding multi-task clustering (MRCMTC) method. They first identify the related instances from the source tasks for each target task, then construct the similarity matrix for each target task by exploiting the related instances from the source tasks based on the Shared Nearest Neighbor similarity, finally perform the spectral clustering on the constructed similarity matrix.
Avory Bryant, Krzysztof Cios	2018	IEEE Transactions on knowledge and data engineering	RNN-DBSCAN: A Density-based Clustering Algorithm using Reverse Nearest Neighbor Density Estimates	In this paper, the authors have proposed a novel density-based clustering algorithm, RNN-DBSCAN, was presented using reverse nearest neighbor based core observation and observation reachability definitions. The superiority of RNN-DBSCAN over prior reverse nearest neighbor approaches, with respect to ARI and NMI performance, was shown using several artificial and real-world datasets. RNN-DBSCAN performance was also shown to be comparable to that of DBSCAN.
Dong Huang, Chang-Dong Wang, Jian-Sheng Wu, Jian-Huang Lai , Chee-Keong Kwoh	2020	IEEE Transactions on knowledge and data engineering	Ultra-Scalable Spectral Clustering and Ensemble Clustering	In this paper, the author proposed two large-scale clustering algorithms, termed ultra-scalable spectral clustering and ultra-scalable ensemble clustering, respectively. In U-SPEC, a new hybrid representative selection strategy is designed to strike a balance between the efficiency of random selection and

Seda Kaya	2020	IEEE Transactions on knowledge and data engineering	Performance Comparison of Supervised Machine Learning Algorithms Before and After PCA and LDA Application: Breast Cancer Detection	the accuracy values for all our models were increased after applying linear discriminant analysis and a success rate of 96.49% was achieved with Logistic Regression.	To predict breast cancer, authors used Wisconsin Diagnostic dataset [8] collected from the UCI Machine Learning Repository. There were 699 instances with total of eleven features	Wisconsin Breast Cancer Dataset
Nahid Nafisi	2018	IEEE Transactions on knowledge and data engineering	Breast Cancer Relapse Prognosis by Classic and Modern Structures of Machine Learning Algorithms	Various machine learning structures are considered to determine whether breast cancer will be relapsed or not. According to, machine learning algorithms can rise the prediction of breast cancer recurrence 15-25%	To predict breast cancer, authors used Wisconsin Diagnostic dataset [8] collected from the UCI Machine Learning Repository. There were 699 instances with total of eleven features	Wisconsin Breast Cancer Dataset
Anusha Bharat	2020	IEEE Transactions on knowledge and data engineering	Using Machine Learning algorithms for breast cancer risk prediction and diagnosis	Not all physicians are experts in distinguishing between benign and malignant tumours and the classification of tumor cells may take up to 2 days. Machine learning algorithms are used to predict the type of cancerous cells	In this research, two machine learning algorithms namely Decision Tree Classifier and Logistic Regression is implemented for prediction of breast cancer, and compared the accuracies of both to find which is best	Wisconsin Breast Cancer Dataset

Process Flow of Model:

STEP 1	Pre-Processing Data
STEP 2	Data Preparation
STEP 3	Features Selection
STEP 4	Feature Projection
STEP 5	Feature Scaling
STEP 6	Model Selection
STEP 7	Prediction



Step 1: PRE-PROCESSING

A data mining approach entitled data pre-processing entails putting original data together into comprehensible format. Actual data is commonly insufficient, unreliable, and prone to various inconsistencies. A tried-and-true methodology for overcoming such conflicts is data initialisation. original data is prepared for subsequent distilling by data initialisation. The UCI information has been pre-processed by using standardized approach. This stage is crucial because the calibre and amount of data individuals acquire will greatly impact how effective your prediction model will be.

Step 2: DATA PREPERATION

Data loading and processing is the process by which we store our info into an appropriate database and make it prepared to be included in machine learning testing. All of our information will be loaded initially, and the arrangement will be randomly done after that.

Step 3: SELECTING FEATURES

Feature selecting, often referred to as variable choosing and attribute selection, is the way of choosing a subcategory of pertinent characteristics to be used in model building in statistical modelling and machine learning. For the choice of the features, we utilised the wrapper method.

Step 4: PROJECTING FEATURES

Data from a relatively high region is transmogrified into a way lower region using feature projection. Depending on the nature of correlations between the available features, both sequential and nonlinear mitigating measures can be applied.

Step 5: FEATURES SCALING

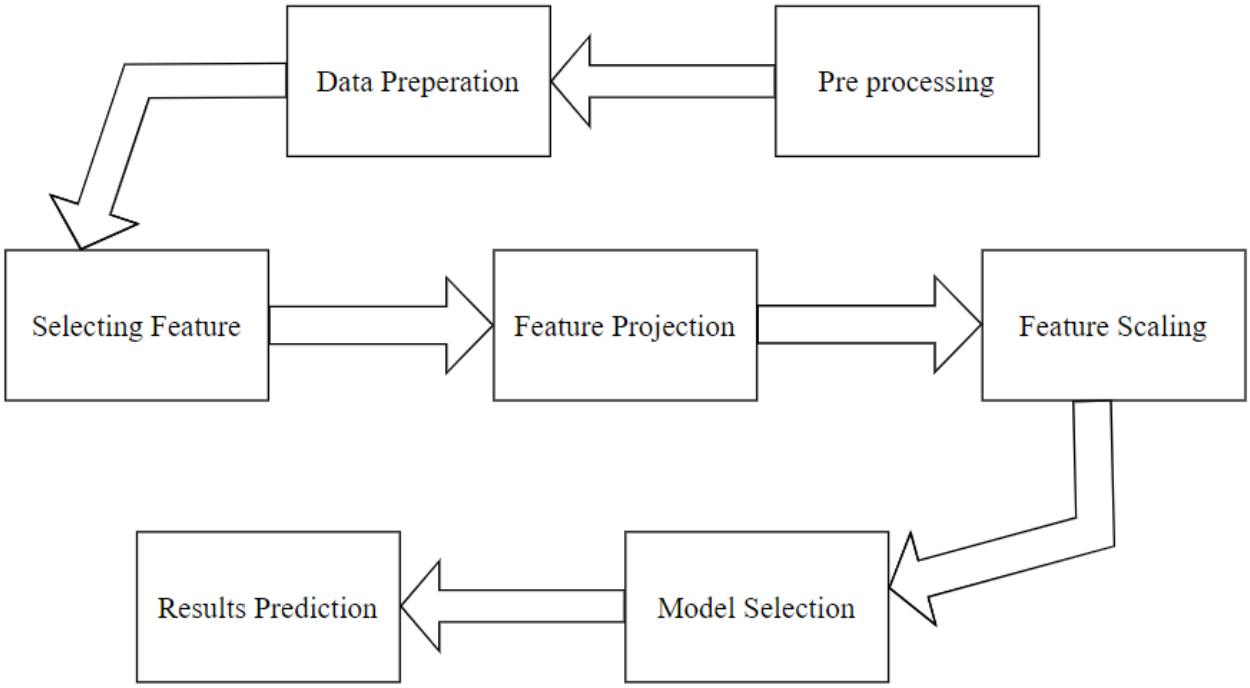
A good set of data will often include characteristics with greater magnitude, units, and span variation. However, because Euclidian distance among two variables has become the most common method used by machine learning techniques, we must adjust the amplitude of each characteristic. Scaling is one possible way to do this.

Step 6: SELECTION OF APPROPRIATE MODEL

Model selection is an important phase in this research because, without choosing an appropriate model, we cannot get results. While there are many models to select from, we choose a model based on our requirements and availability.

Step 7: OUTCOME PREDICTION

Metadata is used by machine learning to offer responses to queries. This is the culmination of all this effort, and this is where machine learning truly shines.



IMPLEMENTATION

In breast cancer it is not necessary that symptoms will be show every time thus helping by taking proper precautions. So, early detection and its proper classification is the only way to lessen the cancer fatality and it is a major task in medical field.

The fundamental problems like ineffectiveness in capturing textural information as well as low retrieval performance caused by poor discrimination of capabilities of features.

Mammography is being used for early-stage detection diagnosis and screening. Key elements here are processing and analysis for better prognosis results[8]. Using FCM technique, image segmentation is performed here. Further, certain features are extracted through these segmented images and trained. Now, the trained images are being classified by an efficient and accurate classifier[15].

Additionally, several characteristics are retrieved using these segmented images. Every data point with in FCM method, sometimes referred to as Fuzzy-C-means, is a member of several clusters that differ in membership level and are dependent on objective function. In order to look into the segmented region, the multileveled Discrete Wavelet Transformation is now being implemented. Matrix-based databases are used to store textual information like cancer cells and pixel values.

This proposed approach aims to classify the images as Initial Stage, Harmful, or Normal after extraction of features and training. It does this by applying the KNN algorithm, which labels it in accordance with the structure of cancer cells. Our system includes the detected boundaries along with the tumor cell after conducting a particular morphological operation and provides clear region attributes, such as Area, Euler Number, and so on. For the extraction of textural characteristics, a wide range of techniques are performed, including PCA, GLCM, and Multi leveled Discrete Wavelet Transform. It is correctly noted and displayed to the doctor where the tumor's area lies.

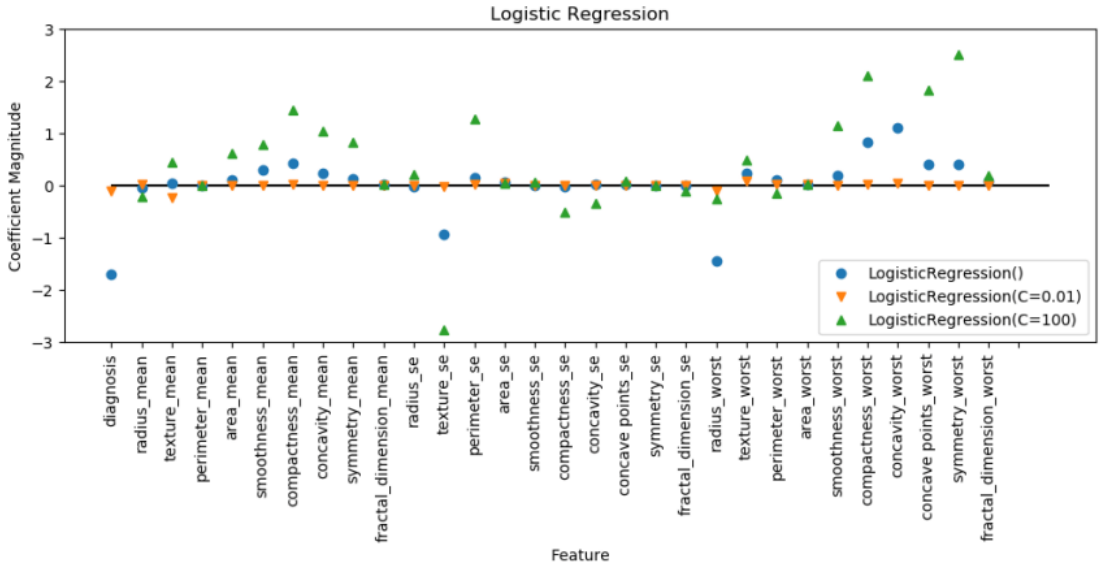
Experimental Investigations

This Project can be done in total of 5 ways which are given below. After Testing each method, we came to know which method gives most accurate outcome.

- Logistic Regression
- K-Nearest Neighbour (KNN)
- Support Vector Machine
- Random and Rotation Forest
- Decision Tree

Logistic Regression:

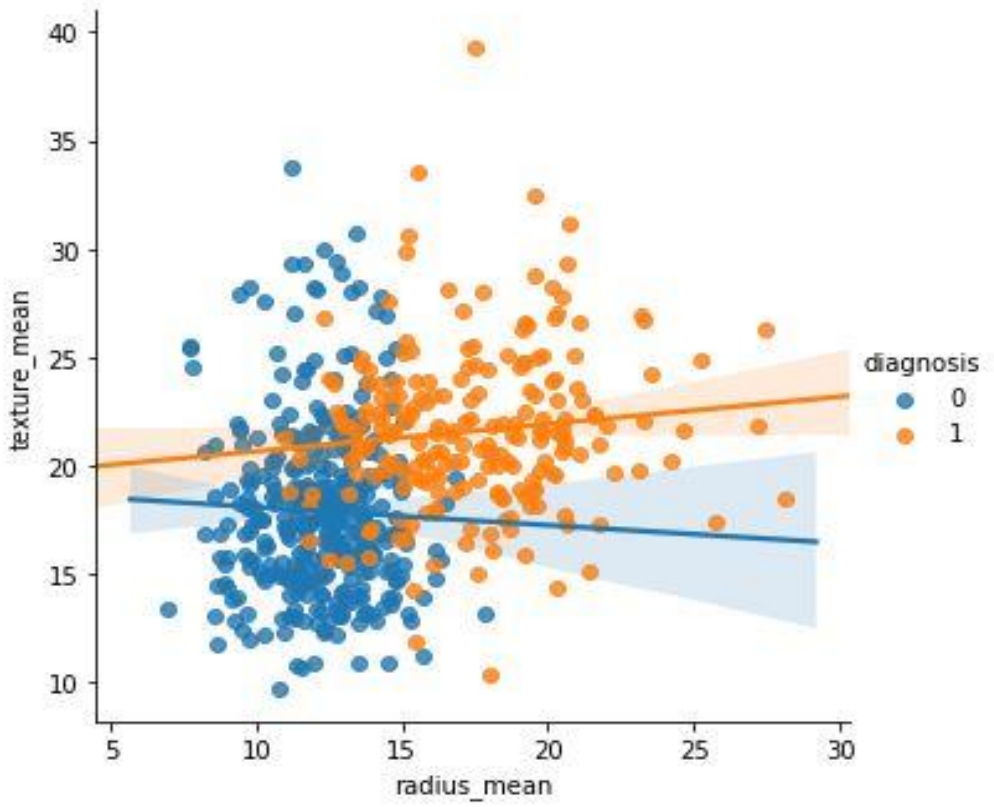
Logistic regression is a technique that firstly used for biological studies in the early twentieth century. It has become widespread for social studies too. Logistic regression is also one of the predictive analyses. Logistic regression is appropriate to use when there is one binary dependent variable and other independent variables. Linear and logistic regressions are different in terms of the dependent variable[18]. Linear regression is a more appropriate technique for continuous variables



K-Nearest Neighbour (KNN):

KNN is a supervised learning technique that means the label of the data is identified before making predictions. Clustering and regression are two purposes to use it. K represents a numerical value for the nearest neighbours. KNN algorithm does not have a training phase. Predictions are made based on the Euclidean distance to k-nearest neighbours[12].

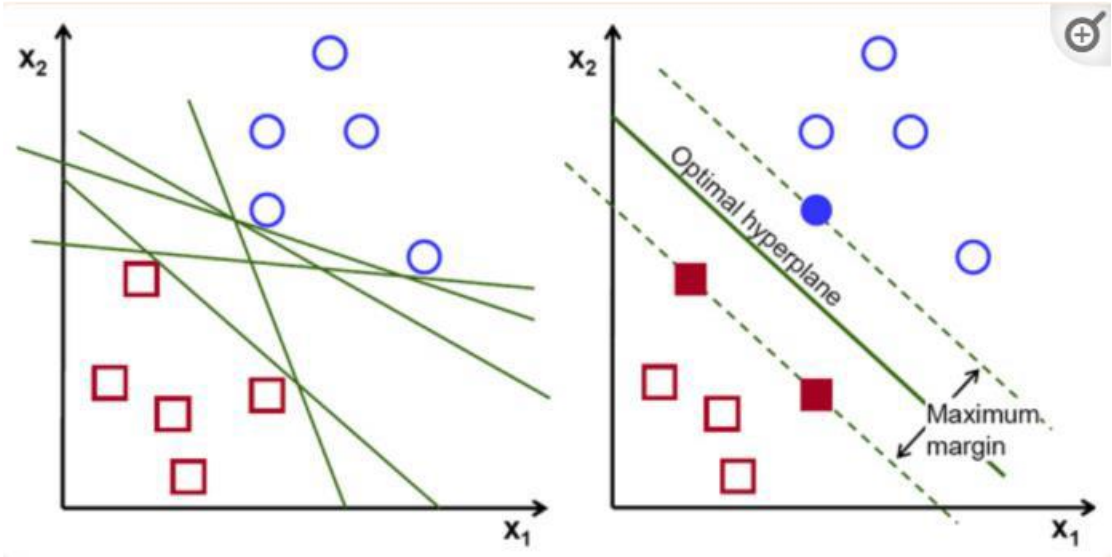
This technique is applied to the prediction of breast cancer dataset since it already has labels such as malignant and benign. The label is classified according to the nearest neighbour to the class labels of its neighbours.



Support Vector Machine:

Support vector machine is one of the most common machine learning techniques. The objective of the algorithm is to find a hyperplane in N-dimensions that classifies the data points. The major part of this algorithm is finding the plane that maximizes the margin[19].

N dimension diversifies based on the feature numbers. Comparing two features could be done smoothly. However, if there are several features for classification, it is not always that straightforward. Maximizing the margin provides more accurate prediction results

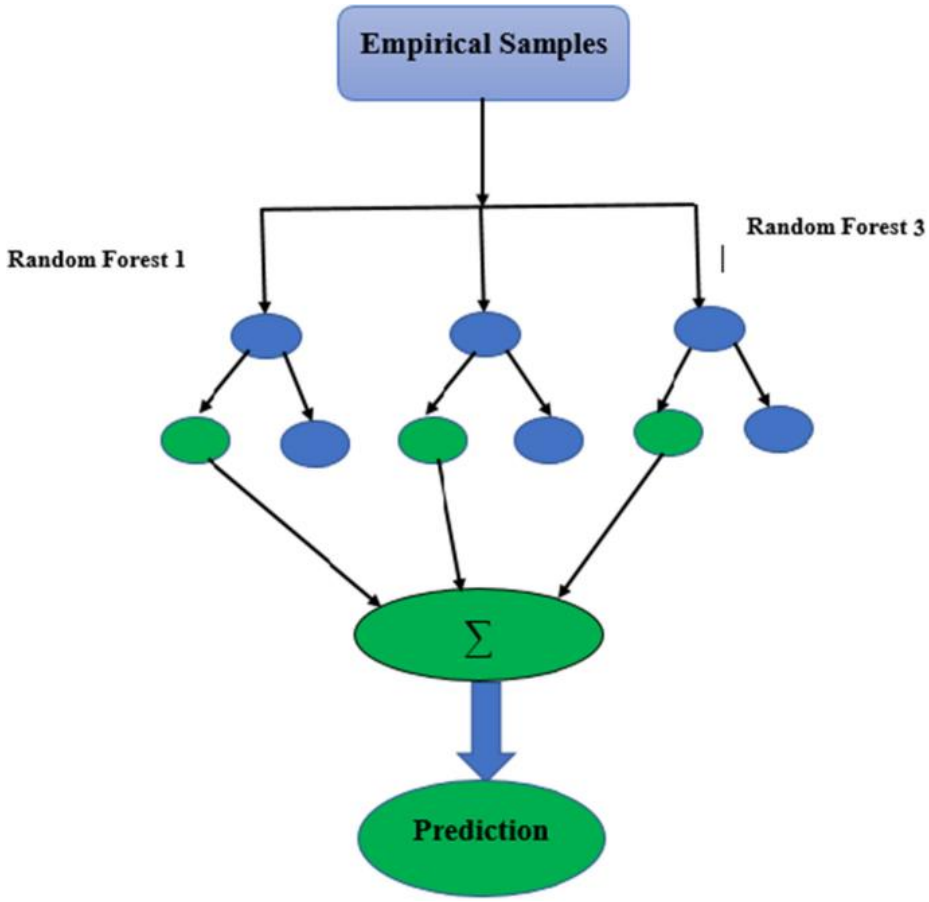


SVM has a small trade off between large margin and accurate classification. If the exact classification without sacrificing any individual sample is applied, the margin could be very narrow, which could lead to a lower accuracy level[10]. On the other hand, by maximizing the margin between classes to get a better accuracy, support vectors that are closest to the hyperplane could be considered with other class members.

Random and Rotation Forest:

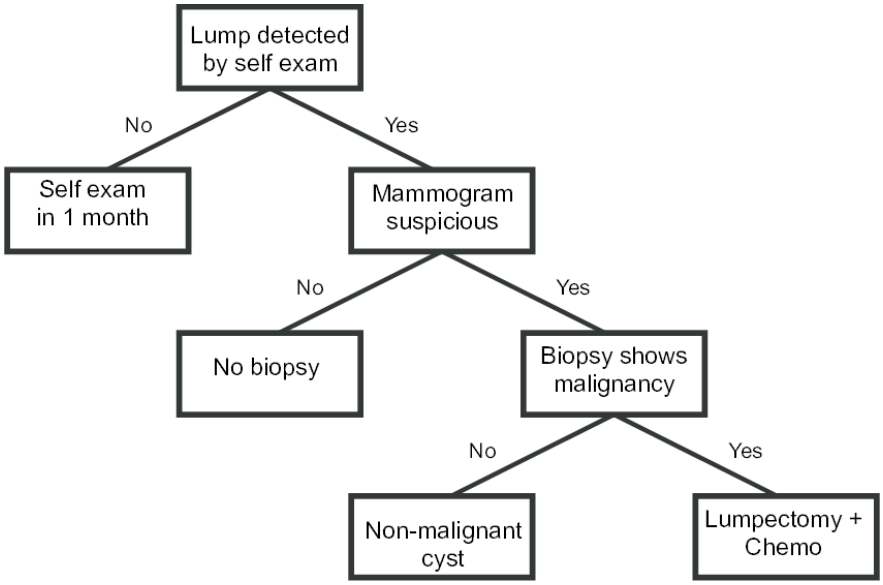
Random forest is an ensemble learning model that can be used for both regression and classification. Indeed, a random forest consists of many decision trees. Therefore, in some cases, it is more logical to use random forest rather than a decision tree[17].

The rotation forest algorithm consists of generating a classifier that is based on the extraction of attributes[20]. The attribute set is randomly grouped into K different subsets. It aims to create accurate and significant classifiers



Decision Tree:

A decision tree (DT) is one of the most common supervised learning techniques. Regression and classification are two main goals to use it. It seeks to solve problems by drawing a tree figure. Features are known as decision nodes, and outputs are leaf nodes. Feature values are considered as categorical in the decision tree algorithm. At the very beginning of this algorithm[11], it is essential to choose the best attribute and place it at the top on tree figure and then split the tree.



After testing all above methods we came to conclusion that usage of Support Vector Machine method, we get more accurate solution which helps in identifying cancer stage as early as possible

Process:

- Information collection from unstructured data
- Conversion of collected data to structured form
- Pattern identification from structured data
- Result analysis
- Extraction and storage

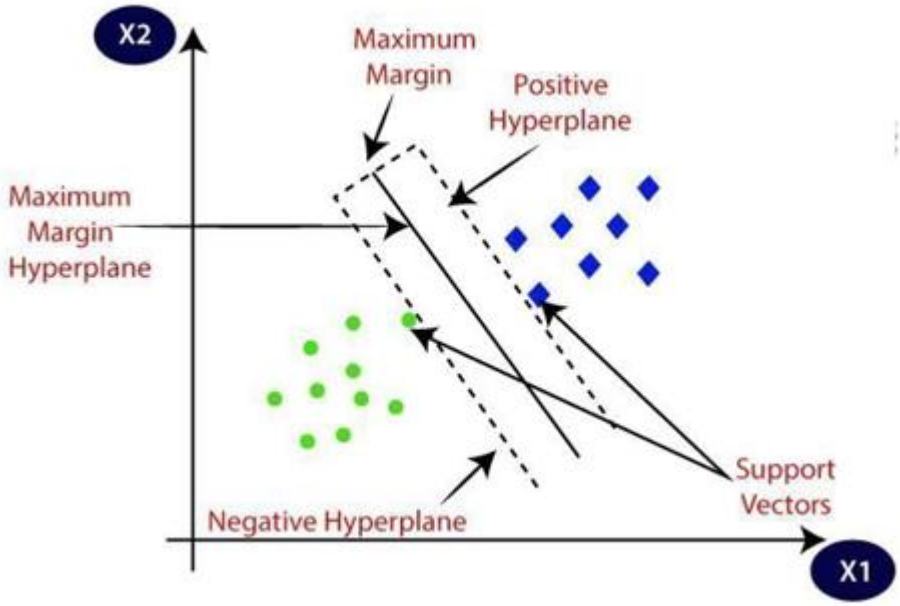
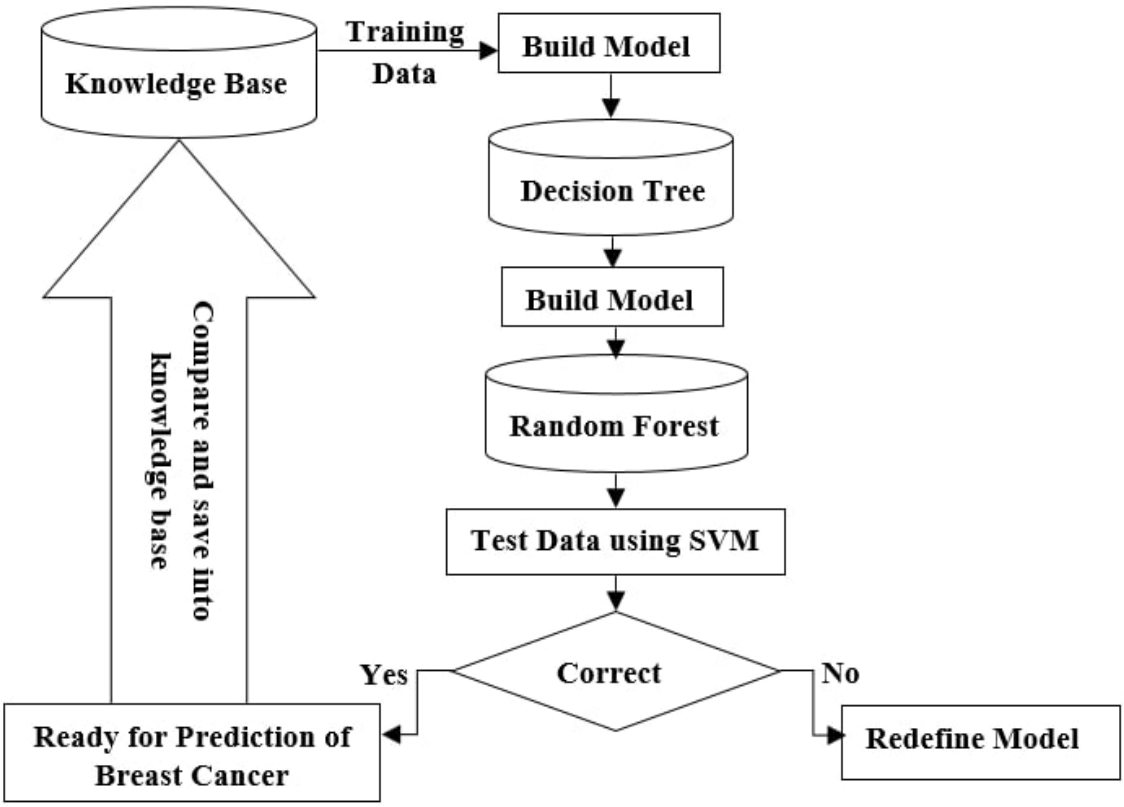


Fig. Support Vector Machine Model

Flow Diagram:



Benefits of using Data science in This Process:

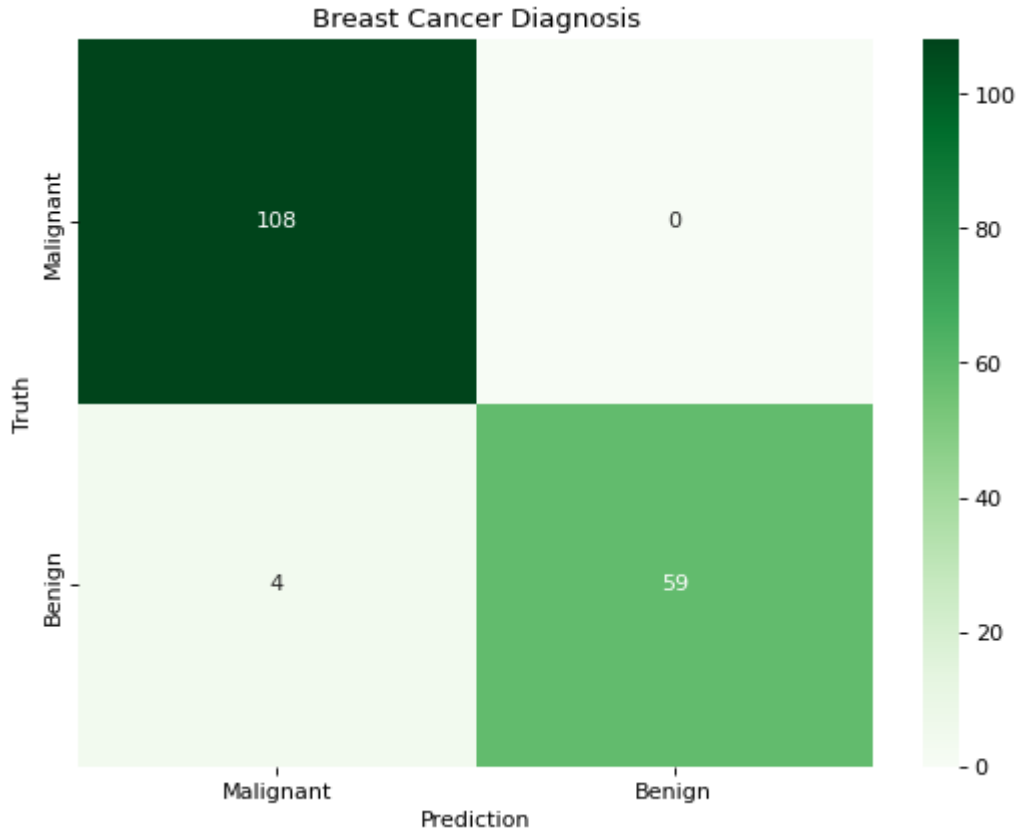
GETTING DATA TO ONE PLACE MAKES IT EASIER TO ANALYSE

USING MULTIPLE CRITERIA CREATES THE CORRECT CONTEXT.

HAVING A BROADER PERSPECTIVE MAKES UNKNOWN RISKS VISIBLE.

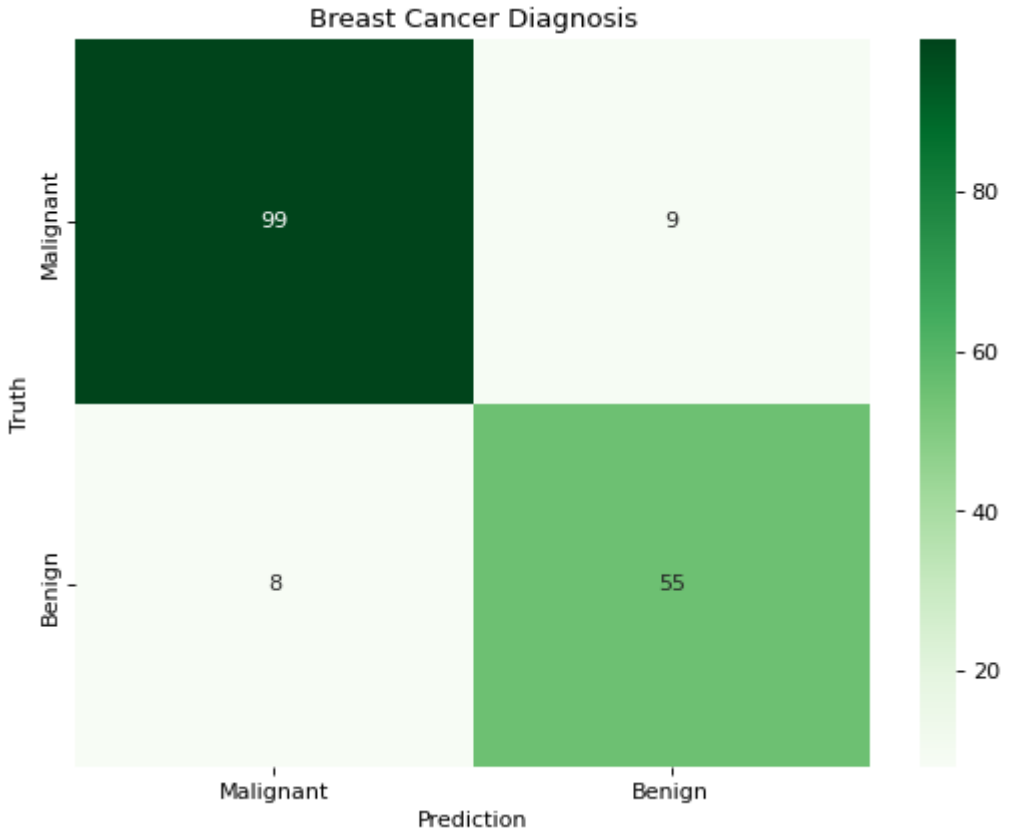
Experimental Results:

Output after Using Support Vector Machine Method:



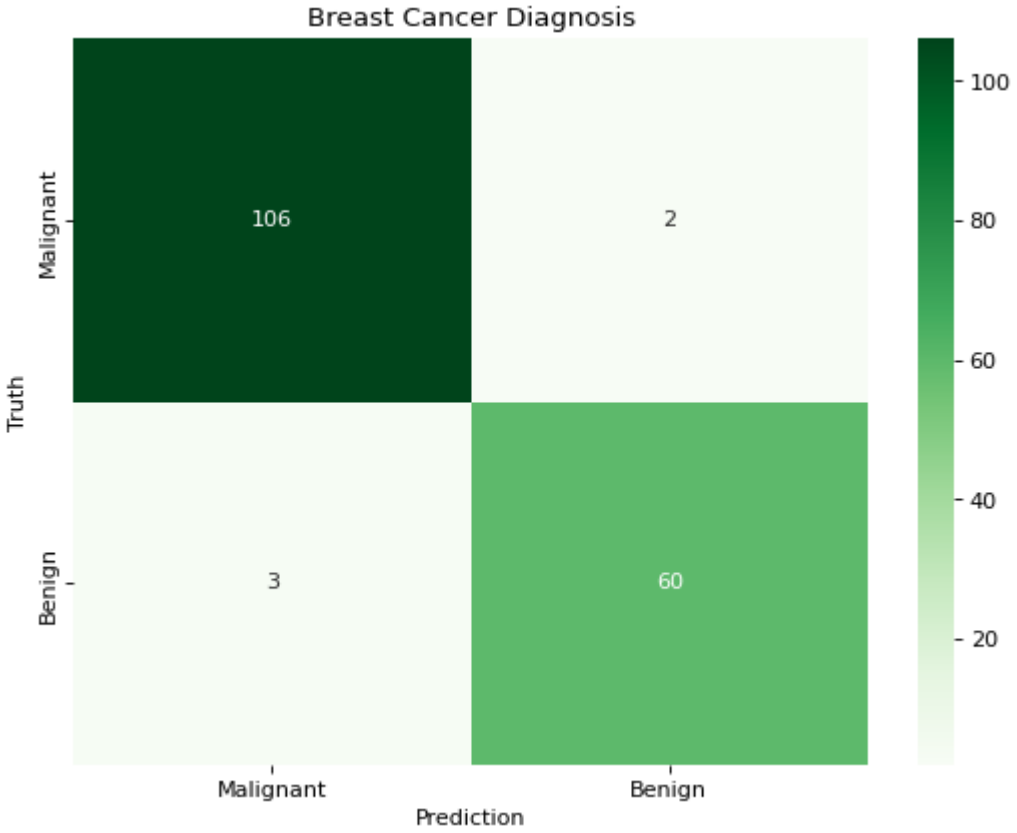

```
Accuracy of Support Vector Classifier: 97.6608187134503

Accuracy of Support Vector Classifier in training set: 98.49246231155779

              precision    recall  f1-score   support

           0       0.96      1.00      0.98       108
           1       1.00      0.94      0.97        63

    accuracy                           0.98       171
   macro avg       0.98      0.97      0.97       171
weighted avg       0.98      0.98      0.98       171
```

Output after Using Naive Bayes model:




```
Accuracy of Naive Bayes model: 90.05847953216374

              precision    recall  f1-score   support

           0       0.93      0.92      0.92       108
           1       0.86      0.87      0.87        63

    accuracy                           0.90       171
   macro avg       0.89      0.89      0.89       171
weighted avg       0.90      0.90      0.90       171
```

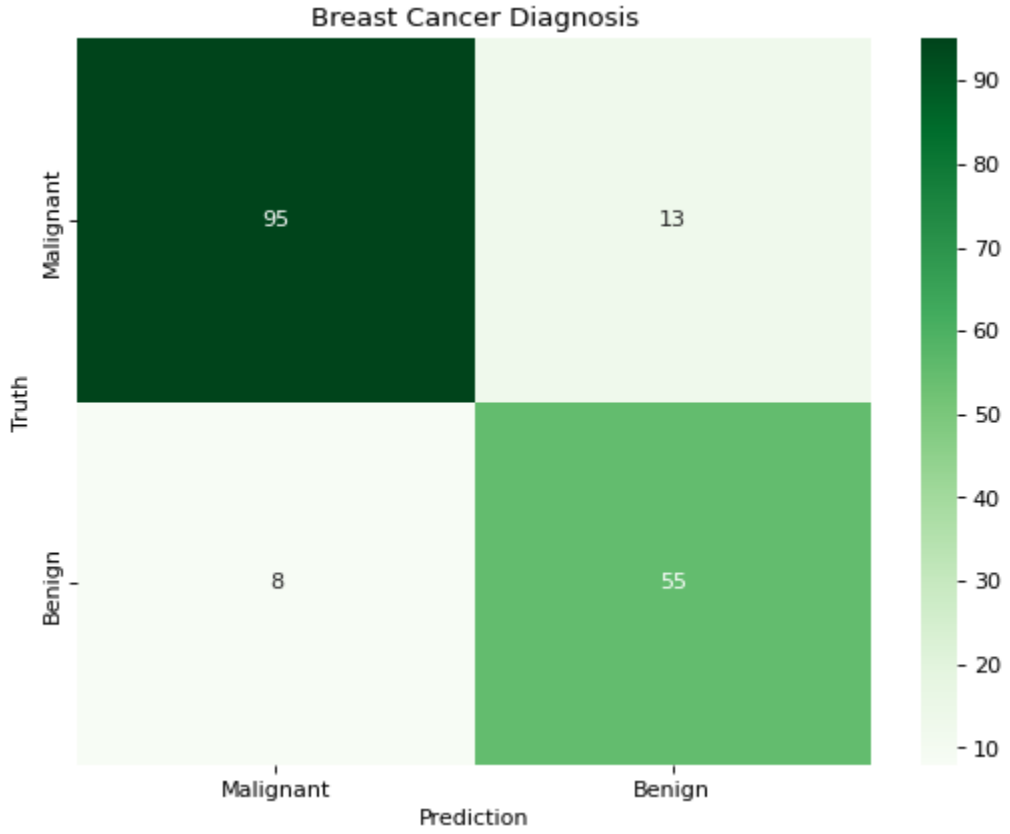
Output after Using Logistic Regression model:



Accuracy of Logistic Regression: 97.07602339181285

	precision	recall	f1-score	support
0	0.97	0.98	0.98	108
1	0.97	0.95	0.96	63
accuracy			0.97	171
macro avg	0.97	0.97	0.97	171
weighted avg	0.97	0.97	0.97	171

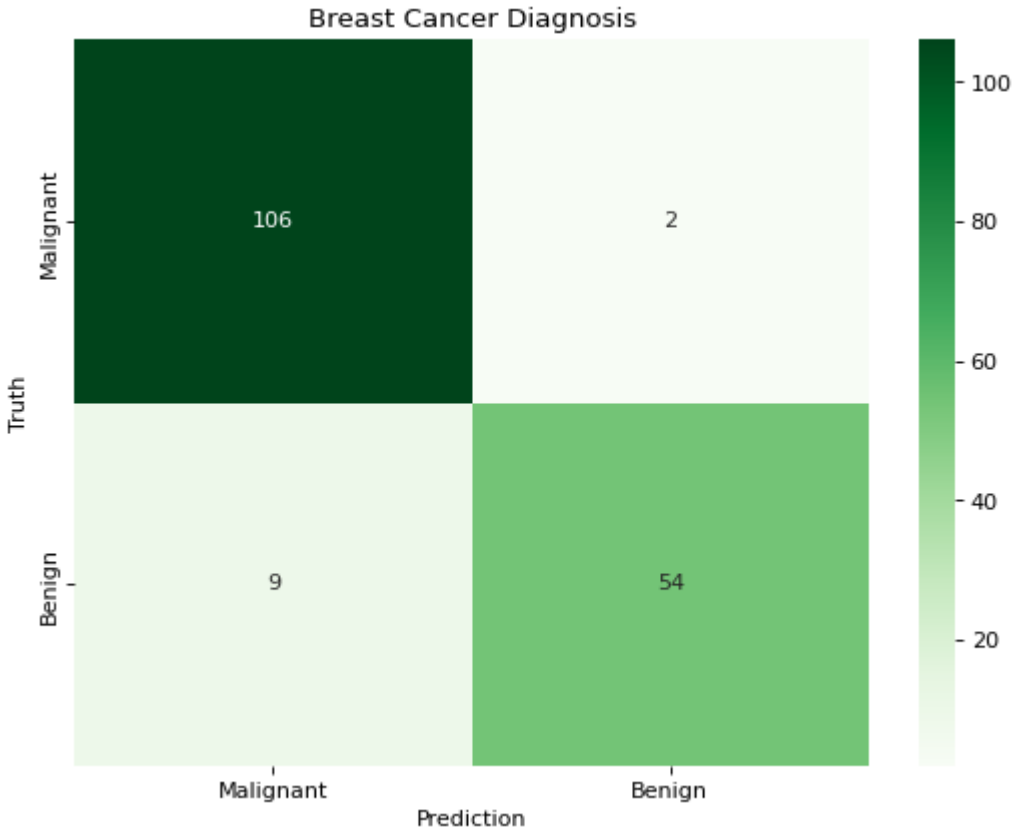
Output after Using Random Forest model:



Accuracy of Random Forest: 87.71929824561403

	precision	recall	f1-score	support
0	0.92	0.88	0.90	108
1	0.81	0.87	0.84	63
accuracy			0.88	171
macro avg	0.87	0.88	0.87	171
weighted avg	0.88	0.88	0.88	171

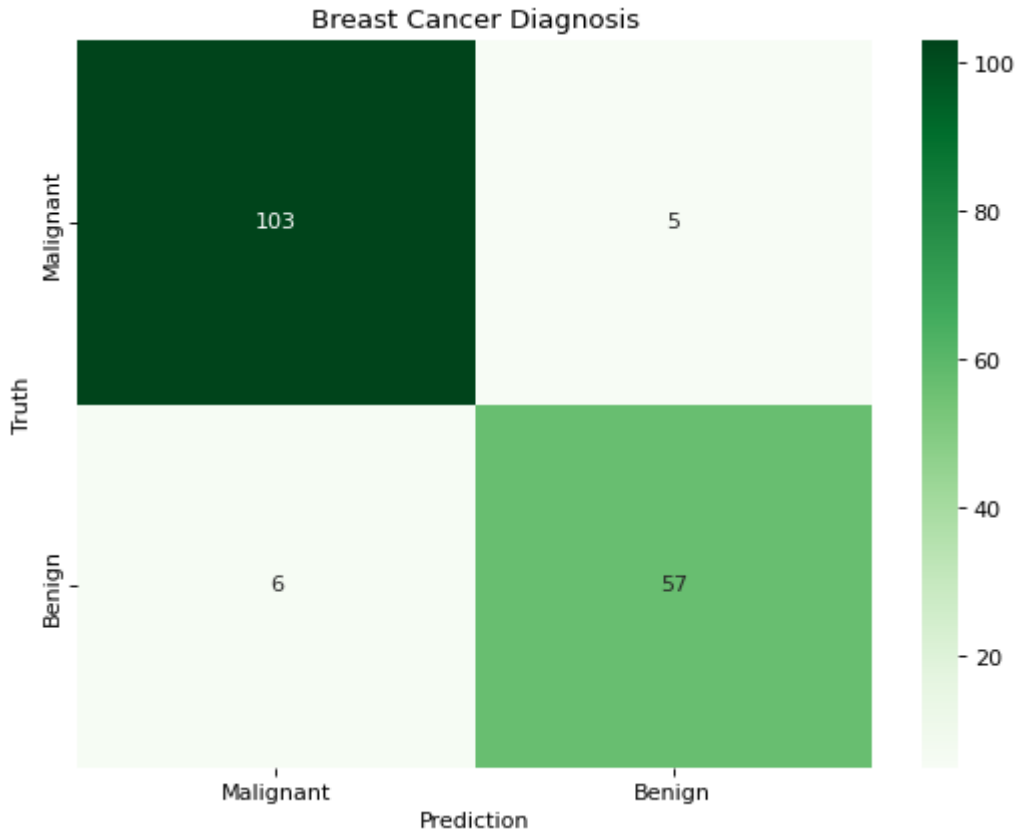
Output after Using K-Neighbors classifier model:



Accuracy of K-NeighborsClassifier: 93.56725146198829

	precision	recall	f1-score	support
0	0.92	0.98	0.95	108
1	0.96	0.86	0.91	63
accuracy			0.94	171
macro avg	0.94	0.92	0.93	171
weighted avg	0.94	0.94	0.93	171

Output after Using Decision Tree model:



Accuracy of DecisionTreeClassifier: 93.56725146198829

	precision	recall	f1-score	support
0	0.94	0.95	0.95	108
1	0.92	0.90	0.91	63
accuracy			0.94	171
macro avg	0.93	0.93	0.93	171
weighted avg	0.94	0.94	0.94	171

Confusion matrix:

A Confusion Matrix is generated for grading classification tasks when the output may use two or more different types of classes. A confusion matrix is a table with the entries "Actual" and "Predicted"

	Predicted **No**	Predicted **Yes**
Actual **No**	**TP** True Positive	**FN** False Negative
Actual **Yes**	**FP** False Positive	**TN** True Negative

Formulas:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN}$$

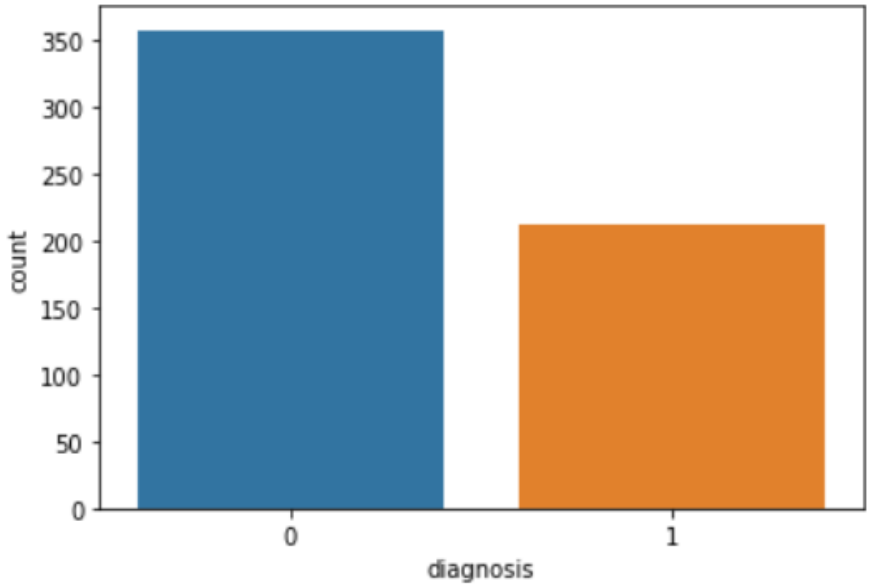
$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

After reading dataset, visualized using bar graph

```
Number of Benign:  357
Number of Malignant :  212
```



So, there are 357 benign cases and 212 malignant cases based on taken data.

We considered 11 values:

```
id
diagnosis
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
```

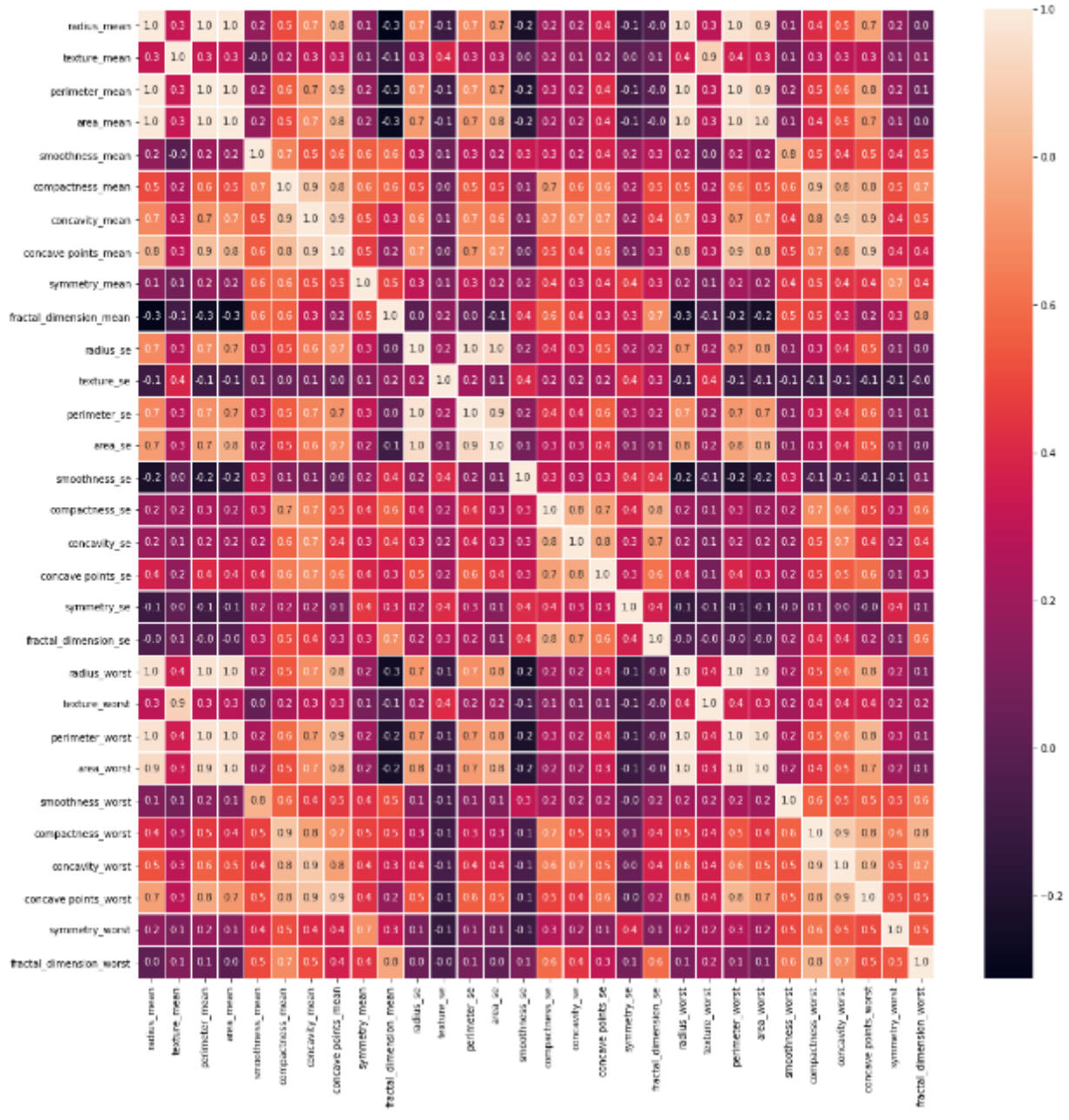
Algorithms	Accuracy Training	Accuracy testing
SVM	98.4	97.2
Random Forest	99.8	96.5
Logistic Regression	95.5	95.8
Decision Tree	98.8	95.1
KNN	94.6	93.7

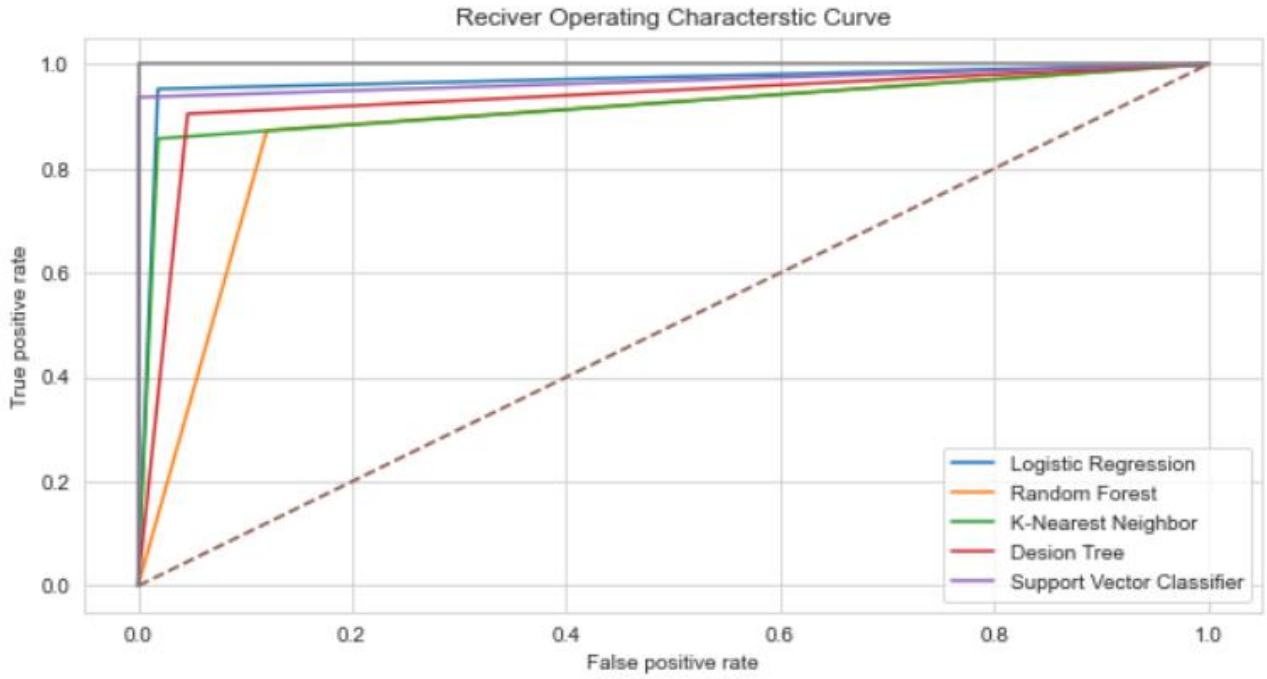
Accuracy %

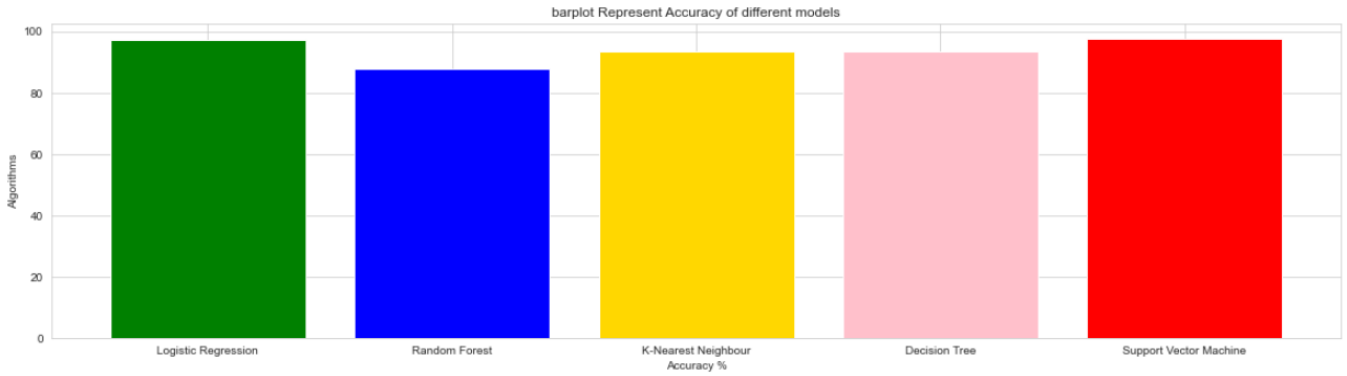
Algorithms	AUC %
SVM	0.966
Random Forest	0.960
Logistic Regression	0.947
Decision Tree	0.945
KNN	0.952

ROC Curve Area

Out[8]: <AxesSubplot:>




Reciver Operating Characterstic Curve
True positive rate
False positive rate
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression
Random Forest
K-Nearest Neighbor
Desion Tree
Support Vector Classifier


barplot Represent Accuracy of different models
Algorithms
Accuracy %
0
20
40
60
80
100
Logistic Regression
Random Forest
K-Nearest Neighbour
Decision Tree
Support Vector Machine

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 16)]              0
_________________________________________________________________
dense (Dense)                (None, 256)               4352
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
reshape (Reshape)            (None, 256, 1)            0
_________________________________________________________________
conv1d (Conv1D)              (None, 256, 16)           80
_________________________________________________________________
batch_normalization (BatchNo (None, 256, 16)           64
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 256, 16)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 64, 16)            1040
_________________________________________________________________
batch_normalization_1 (Batch (None, 64, 16)            64
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 64, 16)            0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 64, 64)            4160
_________________________________________________________________
batch_normalization_2 (Batch (None, 64, 64)            256
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 64, 64)            0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 16, 64)            16448
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 64)            256
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 16, 64)            0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 8, 64)             16448
_________________________________________________________________
batch_normalization_4 (Batch (None, 8, 64)             256
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 8, 64)             0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               65664
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                4128
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 375,905
Trainable params: 375,457
Non-trainable params: 448
```

Code:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sn
from collections import Counter
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

```
from sklearn.metrics import
confusion_matrix,accuracy_score,roc_curve,classification_report
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC

data = pd.read_csv("data.csv")
data.head()

data.info()

list = ['Unnamed: 32','id','diagnosis']
x = data.drop(list,axis = 1 )
print(data['diagnosis'])
data['diagnosis'] = data['diagnosis'].replace(['M'],1)
data['diagnosis'] = data['diagnosis'].replace(['B'],0)
y = data.diagnosis

ax = sn.countplot(y,label="Count")
B, M = y.value_counts()
print('Number of Benign: ',B)
print('Number of Malignant : ',M)

x.describe()


x.head()

f,ax = plt.subplots(figsize=(18, 18))
sn.heatmap(x.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)

drop_list1 = ['perimeter_mean','radius_mean','compactness_mean','concave
points_mean','radius_se','perimeter_se','radius_worst','perimeter_worst','compac
tness_worst','concave points_worst','compactness_se','concave
points_se','texture_worst','area_worst']
x_1 = x.drop(drop_list1,axis = 1 )
x_1.head()
```

```
f,ax = plt.subplots(figsize=(14, 14))
sn.heatmap(x_1.corr(), annot=True, linewidths=.5, fmt= '.1f',ax=ax)

x_1.columns

X_train, X_test, y_train, y_test = train_test_split(x_1, y, test_size=0.30,
random_state = 0)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)


m7 = 'Support Vector Classifier'
svc =  SVC(kernel='rbf', C=2)
svc.fit(X_train, y_train)
svc_predicted = svc.predict(X_test)
svc_conf_matrix = confusion_matrix(y_test, svc_predicted)
svc_acc_score = accuracy_score(y_test, svc_predicted)
svc_predictedtrain = svc.predict(X_train)
svc_acc_scoretrain = accuracy_score(y_train, svc_predictedtrain)

plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(svc_conf_matrix, cmap="Greens", annot=True, fmt='d',
xticklabels= [ 'Malignant','Benign'],
            yticklabels=['Malignant', 'Benign'])
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print("Accuracy of Support Vector Classifier:",svc_acc_score*100,'\n')
print("Accuracy of Support Vector Classifier in training
set:",svc_acc_scoretrain*100,'\n')

print(classification_report(y_test,svc_predicted))

m2 = 'Naive Bayes'
nb = GaussianNB()
nb.fit(X_train,y_train)
```

```
nbpred = nb.predict(X_test)
nb_conf_matrix = confusion_matrix(y_test, nbpred)
nb_acc_score = accuracy_score(y_test, nbpred)
plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(nb_conf_matrix, cmap="Greens", annot=True, fmt='d',
xticklabels= [ 'Malignant','Benign'],
            yticklabels=['Malignant', 'Benign'])
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print("Accuracy of Naive Bayes model:",nb_acc_score*100,'\n')
print(classification_report(y_test,nbpred))


import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.models import Sequential

features_inputs = tf.keras.Input((16, ), dtype=tf.float16)

feature_x = layers.Dense(256, activation='relu')(features_inputs)
feature_x = layers.Dropout(0.1)(feature_x)
feature_x = layers.Reshape((-1,1))(feature_x)
feature_x = layers.Conv1D(filters=16, kernel_size=4, strides=1,
padding='same')(feature_x)
feature_x = layers.BatchNormalization()(feature_x)
feature_x = layers.LeakyReLU()(feature_x)
feature_x = layers.Conv1D(filters=16, kernel_size=4, strides=4,
padding='same')(feature_x)
feature_x = layers.BatchNormalization()(feature_x)
feature_x = layers.LeakyReLU()(feature_x)
feature_x = layers.Conv1D(filters=64, kernel_size=4, strides=1,
padding='same')(feature_x)
feature_x = layers.BatchNormalization()(feature_x)
feature_x = layers.LeakyReLU()(feature_x)
feature_x = layers.Conv1D(filters=64, kernel_size=4, strides=4,
padding='same')(feature_x)
feature_x = layers.BatchNormalization()(feature_x)
feature_x = layers.LeakyReLU()(feature_x)
```

```
feature_x = layers.Conv1D(filters=64, kernel_size=4, strides=2,
padding='same')(feature_x)
feature_x = layers.BatchNormalization()(feature_x)
feature_x = layers.LeakyReLU()(feature_x)
feature_x = layers.Flatten()(feature_x)
```

```
_________________________________________________________________
Epoch 1/100
13/13 [==============================] - 4s 53ms/step - loss: 8.2271 - accuracy: 0.8518 - val_loss: 8.4710 - val_accuracy: 0.63
16
Epoch 2/100
13/13 [==============================] - 0s 18ms/step - loss: 7.4961 - accuracy: 0.9296 - val_loss: 7.5684 - val_accuracy: 0.63
16
Epoch 3/100
13/13 [==============================] - 0s 17ms/step - loss: 6.8589 - accuracy: 0.9623 - val_loss: 7.1954 - val_accuracy: 0.63
16
Epoch 4/100
13/13 [==============================] - 0s 16ms/step - loss: 6.1561 - accuracy: 0.9749 - val_loss: 6.7095 - val_accuracy: 0.63
16
Epoch 5/100
13/13 [==============================] - 0s 17ms/step - loss: 5.7240 - accuracy: 0.9698 - val_loss: 5.8149 - val_accuracy: 0.63
16
Epoch 6/100
13/13 [==============================] - 0s 16ms/step - loss: 5.1994 - accuracy: 0.9724 - val_loss: 5.5487 - val_accuracy: 0.64
91
Epoch 7/100
13/13 [==============================] - 0s 15ms/step - loss: 4.9093 - accuracy: 0.9698 - val_loss: 5.0662 - val_accuracy: 0.64
91
Epoch 8/100
13/13 [==============================] - 0s 17ms/step - loss: 4.4354 - accuracy: 0.9749 - val_loss: 4.6084 - val_accuracy: 0.68
42
Epoch 9/100
13/13 [==============================] - 0s 16ms/step - loss: 4.0515 - accuracy: 0.9824 - val_loss: 4.2583 - val_accuracy: 0.70
76
Epoch 10/100
13/13 [==============================] - 0s 14ms/step - loss: 3.7746 - accuracy: 0.9874 - val_loss: 3.9065 - val_accuracy: 0.81
29
Epoch 11/100
13/13 [==============================] - 0s 14ms/step - loss: 3.4541 - accuracy: 0.9925 - val_loss: 3.5646 - val_accuracy: 0.84
21
Epoch 12/100
13/13 [==============================] - 0s 14ms/step - loss: 3.1862 - accuracy: 0.9925 - val_loss: 3.2547 - val_accuracy: 0.92
40
Epoch 13/100
13/13 [==============================] - 0s 15ms/step - loss: 2.9925 - accuracy: 0.9899 - val_loss: 3.2080 - val_accuracy: 0.84
21
Epoch 14/100
13/13 [==============================] - 0s 14ms/step - loss: 2.7860 - accuracy: 0.9824 - val_loss: 2.7860 - val_accuracy: 0.92
40
Epoch 15/100
13/13 [==============================] - 0s 14ms/step - loss: 2.5486 - accuracy: 0.9899 - val_loss: 2.6214 - val_accuracy: 0.95
91
Epoch 16/100
13/13 [==============================] - 0s 14ms/step - loss: 2.3654 - accuracy: 0.9899 - val_loss: 2.5273 - val_accuracy: 0.88
89
Epoch 17/100
13/13 [==============================] - 0s 14ms/step - loss: 2.2073 - accuracy: 0.9849 - val_loss: 2.2610 - val_accuracy: 0.92
98
Epoch 18/100
13/13 [==============================] - 0s 15ms/step - loss: 2.0603 - accuracy: 0.9899 - val_loss: 2.1065 - val_accuracy: 0.93
57
Epoch 19/100
13/13 [==============================] - 0s 18ms/step - loss: 2.5571 - accuracy: 0.9422 - val_loss: 2.3608 - val_accuracy: 0.90
06
Epoch 20/100
13/13 [==============================] - 0s 17ms/step - loss: 2.1274 - accuracy: 0.9623 - val_loss: 2.3644 - val_accuracy: 0.84
21
Epoch 21/100
13/13 [==============================] - 0s 18ms/step - loss: 2.1448 - accuracy: 0.9623 - val_loss: 1.9014 - val_accuracy: 0.92
98
Epoch 22/100
13/13 [==============================] - 0s 17ms/step - loss: 1.9330 - accuracy: 0.9673 - val_loss: 1.5975 - val_accuracy: 0.95
32
Epoch 23/100
13/13 [==============================] - 0s 16ms/step - loss: 1.7596 - accuracy: 0.9724 - val_loss: 1.6334 - val_accuracy: 0.91
81
Epoch 24/100
13/13 [==============================] - 0s 15ms/step - loss: 1.9906 - accuracy: 0.9472 - val_loss: 2.2486 - val_accuracy: 0.88
89
Epoch 25/100
13/13 [==============================] - 0s 14ms/step - loss: 2.6247 - accuracy: 0.8844 - val_loss: 2.3703 - val_accuracy: 0.87
72
```

```
x = layers.Dense(512, activation='relu', kernel_regularizer="l2")(feature_x)
```

```
x = layers.Dropout(0.1)(x)
x = layers.Dense(128, activation='relu', kernel_regularizer="l2")(x)
x = layers.Dropout(0.1)(x)
x = layers.Dense(32, activation='relu', kernel_regularizer="l2")(x)
x = layers.Dropout(0.1)(x)
output = layers.Dense(1)(x)
model = tf.keras.Model(inputs=[features_inputs], outputs=[output])
model.summary()
model.compile(optimizer='adam',loss='binary_crossentropy',metrics=['accuracy
'])
epochs = 100
history =
model.fit(X_train,y_train,epochs=100,validation_data=(X_test,y_test))
test_loss,test_accuracy = model.evaluate(X_test,y_test)

from tensorflow import keras
model.summary()
keras.utils.plot_model(model, show_shapes=True)


m1 = 'Logistic Regression'
lr = LogisticRegression()
model = lr.fit(X_train, y_train)
lr_predict = lr.predict(X_test)
lr_conf_matrix = confusion_matrix(y_test, lr_predict)
lr_acc_score = accuracy_score(y_test, lr_predict)
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 16)]              0
_________________________________________________________________
dense (Dense)                (None, 256)               4352
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
reshape (Reshape)            (None, 256, 1)            0
_________________________________________________________________
conv1d (Conv1D)              (None, 256, 16)           80
_________________________________________________________________
batch_normalization (BatchNo (None, 256, 16)           64
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 256, 16)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 64, 16)            1040
_________________________________________________________________
batch_normalization_1 (Batch (None, 64, 16)            64
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 64, 16)            0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 64, 64)            4160
_________________________________________________________________
batch_normalization_2 (Batch (None, 64, 64)            256
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 64, 64)            0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 16, 64)            16448
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 64)            256
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 16, 64)            0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 8, 64)             16448
_________________________________________________________________
batch_normalization_4 (Batch (None, 8, 64)             256
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 8, 64)             0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               65664
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                4128
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 375,905
Trainable params: 375,457
Non-trainable params: 448
```

```
plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(lr_conf_matrix, cmap="Greens", annot=True, fmt='d',
xticklabels= [ 'Malignant','Benign'],
            yticklabels=['Malignant', 'Benign'])
```

```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 16)]              0
_________________________________________________________________
dense (Dense)                (None, 256)               4352
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
reshape (Reshape)            (None, 256, 1)            0
_________________________________________________________________
conv1d (Conv1D)              (None, 256, 16)           80
_________________________________________________________________
batch_normalization (BatchNo (None, 256, 16)           64
_________________________________________________________________
leaky_re_lu (LeakyReLU)      (None, 256, 16)           0
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 64, 16)            1040
_________________________________________________________________
batch_normalization_1 (Batch (None, 64, 16)            64
_________________________________________________________________
leaky_re_lu_1 (LeakyReLU)    (None, 64, 16)            0
_________________________________________________________________
conv1d_2 (Conv1D)            (None, 64, 64)            4160
_________________________________________________________________
batch_normalization_2 (Batch (None, 64, 64)            256
_________________________________________________________________
leaky_re_lu_2 (LeakyReLU)    (None, 64, 64)            0
_________________________________________________________________
conv1d_3 (Conv1D)            (None, 16, 64)            16448
_________________________________________________________________
batch_normalization_3 (Batch (None, 16, 64)            256
_________________________________________________________________
leaky_re_lu_3 (LeakyReLU)    (None, 16, 64)            0
_________________________________________________________________
conv1d_4 (Conv1D)            (None, 8, 64)             16448
_________________________________________________________________
batch_normalization_4 (Batch (None, 8, 64)             256
_________________________________________________________________
leaky_re_lu_4 (LeakyReLU)    (None, 8, 64)             0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 128)               65664
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_3 (Dense)              (None, 32)                4128
_________________________________________________________________
dropout_3 (Dropout)          (None, 32)                0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 33
=================================================================
Total params: 375,905
Trainable params: 375,457
Non-trainable params: 448
```

```
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print("Accuracy of Logistic Regression:",lr_acc_score*100,'\n')
print(classification_report(y_test,lr_predict))

predict = model.predict(X_test)
for i in range(len(predict)):
    if(predict[i]>0.5):
        predict[i]=1
    else:
```

```
        predict[i]=0


conf_arr = confusion_matrix(y_test, predict)

plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(conf_arr, cmap="Greens", annot=True, fmt='d', xticklabels= [
'Malignant','Benign'],
                yticklabels=['Malignant', 'Benign'])
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print(classification_report(y_test,predict))


m1 = 'Logistic Regression'
lr = LogisticRegression()
model = lr.fit(X_train, y_train)
lr_predict = lr.predict(X_test)
lr_conf_matrix = confusion_matrix(y_test, lr_predict)
lr_acc_score = accuracy_score(y_test, lr_predict)

plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(lr_conf_matrix, cmap="Greens", annot=True, fmt='d',
xticklabels= [ 'Malignant','Benign'],
                yticklabels=['Malignant', 'Benign'])
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print("Accuracy of Logistic Regression:",lr_acc_score*100,'\n')
print(classification_report(y_test,lr_predict))


m3 = 'Random Forest Classfier'
rf = RandomForestClassifier(n_estimators=4, random_state=12,max_depth=5)
rf.fit(X_train,y_train)
rf_predicted = rf.predict(X_test)
rf_conf_matrix = confusion_matrix(y_test, rf_predicted)
```

```
rf_acc_score = accuracy_score(y_test, rf_predicted)
plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(rf_conf_matrix, cmap="Greens", annot=True, fmt='d',
xticklabels= [ 'Malignant','Benign'],
            yticklabels=['Malignant', 'Benign'])
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print("Accuracy of Random Forest:",rf_acc_score*100,'\n')
print(classification_report(y_test,rf_predicted))

knn2 = KNeighborsClassifier()

param_grid = {'n_neighbors': np.arange(1, 25)}
knn_gscv = GridSearchCV(knn2, param_grid, cv=3)
knn_gscv.fit(x_1, y)
print(knn_gscv.best_params_)
print(knn_gscv.best_score_)

m5 = 'K-NeighborsClassifier'
knn = KNeighborsClassifier(n_neighbors=18)
knn.fit(X_train, y_train)
knn_predicted = knn.predict(X_test)
knn_conf_matrix = confusion_matrix(y_test, knn_predicted)
knn_acc_score = accuracy_score(y_test, knn_predicted)
plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(knn_conf_matrix, cmap="Greens", annot=True, fmt='d',
xticklabels= [ 'Malignant','Benign'],
            yticklabels=['Malignant', 'Benign'])
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print("Accuracy of K-NeighborsClassifier:",knn_acc_score*100,'\n')
print(classification_report(y_test,knn_predicted))

m6 = 'DecisionTreeClassifier'
dt = DecisionTreeClassifier(criterion = 'entropy',random_state=0,max_depth =
6)
```

```
dt.fit(X_train, y_train)
dt_predicted = dt.predict(X_test)
dt_conf_matrix = confusion_matrix(y_test, dt_predicted)
dt_acc_score = accuracy_score(y_test, dt_predicted)
plt.figure(figsize=(8, 6), dpi=80, facecolor='w', edgecolor='k')
ax = sn.heatmap(dt_conf_matrix, cmap="Greens", annot=True, fmt='d',
xticklabels= [ 'Malignant','Benign'],
            yticklabels=['Malignant', 'Benign'])
plt.title('Breast Cancer Diagnosis')
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.show(ax)
print("Accuracy of DecisionTreeClassifier:",dt_acc_score*100,'\n')
print(classification_report(y_test,dt_predicted))

dnn_false_positive_rate,dnn_true_positive_rate,svc_threshold =
roc_curve(y_test,predict)
lr_false_positive_rate,lr_true_positive_rate,lr_threshold =
roc_curve(y_test,lr_predict)
nb_false_positive_rate,nb_true_positive_rate,nb_threshold =
roc_curve(y_test,nbpred)
rf_false_positive_rate,rf_true_positive_rate,rf_threshold =
roc_curve(y_test,rf_predicted)
knn_false_positive_rate,knn_true_positive_rate,knn_threshold =
roc_curve(y_test,knn_predicted)
dt_false_positive_rate,dt_true_positive_rate,dt_threshold =
roc_curve(y_test,dt_predicted)
svc_false_positive_rate,svc_true_positive_rate,svc_threshold =
roc_curve(y_test,svc_predicted)


sn.set_style('whitegrid')
plt.figure(figsize=(10,5))
plt.title('Reciver Operating Characterstic Curve')
plt.plot(lr_false_positive_rate,lr_true_positive_rate,label='Logistic Regression')
plt.plot(rf_false_positive_rate,rf_true_positive_rate,label='Random Forest')
plt.plot(knn_false_positive_rate,knn_true_positive_rate,label='K-Nearest
Neighbor')
plt.plot(dt_false_positive_rate,dt_true_positive_rate,label='Desion Tree')
```

```
plt.plot(svc_false_positive_rate,svc_true_positive_rate,label='Support Vector Classifier')
plt.plot([0,1],ls='--')
plt.plot([0,0],[1,0],c='.5')
plt.plot([1,1],c='.5')
plt.ylabel('True positive rate')
plt.xlabel('False positive rate')
plt.legend()
plt.show()

model_ev = pd.DataFrame({'Model': ['Logistic Regression','Random Forest',
                'K-Nearest Neighbour','Decision Tree','Support Vector Machine'],
'Accuracy': [lr_acc_score*100,

rf_acc_score*100,knn_acc_score*100,dt_acc_score*100,svc_acc_score*100]})
model_ev

colors = ['green','blue','gold','pink','red']
plt.figure(figsize=(20,5))
plt.title("barplot Represent Accuracy of different models")
plt.xlabel("Accuracy %")
plt.ylabel("Algorithms")
plt.bar(model_ev['Model'],model_ev['Accuracy'],color = colors)
plt.show()
```

DISCUSSION OF RESULTS

This Project can be done in total of 5 ways which are given above. After Testing each method, we came to know which method gives most accurate outcome

In this project, logistic regression, k-nearest neighbor, support vector machine, random forest, decision tree, and naïve Bayes classification algorithms were created, and accuracy scores for each of them were obtained. Each algorithm was applied to three different datasets that included various features. The first dataset covered all independent features, the second dataset included highly correlated features[14], and the last dataset included low correlated features. Three different datasets were used separately for each machine learning technique, and accuracy results were obtained to make comparisons. SVM gave better accuracy results rather than the other methods. The main advantage of SVM is that it is very efficient to train. In addition, the SVM model is useful and gives more accurate results in complex algorithms[16]. A competitive performance was demonstrated when dealing with imbalanced data (97.66% accuracy).

Out[26]:

	Model	Accuracy
0	Logistic Regression	97.076023
1	Random Forest	87.719298
2	K-Nearest Neighbour	93.567251
3	Decision Tree	93.567251
4	Support Vector Machine	97.660819

CONCLUSION

In this project in python, we learned to build a breast cancer tumour predictor on the wisconsin dataset and created graphs and results for the same. It has been observed that a good dataset provides better accuracy. Selection of appropriate algorithms with good home dataset will lead to the development of prediction systems. These systems can assist in proper treatment methods for a patient diagnosed with breast cancer. There are many treatments for a patient based on breast cancer stage; data mining and machine learning can be a very good help in deciding the line of treatment to be followed by extracting knowledge from such suitable databases.

In this project, five distinct machine learning techniques were investigated for breast cancer diagnosis. SVM gave better accuracy results rather than the other methods. The main advantage of SVM is that it is very efficient to train. In addition, the SVM model is useful and gives more accurate results in complex algorithms. A competitive performance was demonstrated when dealing with imbalanced data (97.66% accuracy).

Future Prediction Analysis

This study has a number of important strengths. First, we have presented an alternative modelling approach for estimation of breast cancer incidence, thereby enhancing the accuracy of the prediction intervals for future incidence rates. Further, these models will be most useful for modelling and projecting the future trends of other cancers as well, for which there has been very little advancement in treatment and opportunities for prevention, early detection, or both, are few. First, our study lacks inclusion of birth cohort effect, breast cancer subtype information and other important risk factors such as screening and treatment options associated with hormone replacement therapy that may influence age-related changes in incidence. Currently FTS model do not incorporate such effects, but smoothing process used in FTS modelling may reduce the variation attributable to such effects.

It is likely that improving prediction models will require inclusion of additional known risk factors which may play a large role in the surveillance, treatment, and survival outcomes of this disease. Secondly, there is a possibility/limitation that there are more breast cancer cases in the city than described herein. Those could potentially include patients who do not have access to hospitals and/or that are diagnosed at other health facility. It could happen that these patients were the most economically disadvantaged. However, our data is based on the largest possible registries of the city, thus serving as the major source in breast cancer data in Paris, and thereby allowing better estimates of national pattern in the absence of any population-based cancer registry in the country

REFRENCES

[1] J. Sivapriya, A. Kumar, S. Siddarth Sai, and S. Sriram, "Breast cancer prediction using machine learning," International Journal of Recent Technology and Engineering (IJRTE), vol. 8, 2019.

[2] A. Toprak, "Extreme learning machine (elm)-based classification of benign and malignant cells in breast cancer," Medical science monitor: international medical journal of experimental and clinical research, vol. 24, p. 6537, 2018

[3] B. Sahiner, H.-P. Chan, N. Petrick, R. F. Wagner, and L. Hadjiiski, "Feature selection and classifier performance in computer-aided diagnosis: The effect of finite sample size," Medical Physics, vol. 27, pp. 1509– 1522, 2000.

[4] Sohail M.N., Jiadong R., Uba M.M., Irshad M. A comprehensive looks at data mining techniques contributing to medical data growth: A survey of researcher reviews; Proceedings of the 35th IEEE International Conference on Computer Design, ICCD 2017; Boston, MA, USA. 5–8 November 2017; pp. 21–26.

[5] Petri I., Kubicki S., Rezgui Y., Guerriero A., Li H. Optimizing energy efficiency in operating built environment assets through building information modeling: A case study. *Energies.* 2017;10:1167. doi: 10.3390/en10081167.

[6] Liao S.H., Chen Y.J., Deng M.Y. Mining customer knowledge for tourism new product development and customer relationship management. *Expert Syst. Appl.* 2010;37:4212–4223. doi: 10.1016/j.eswa.2009.11.081.

[7] Jothi N., Rashid N.A., Husain W. Data mining in healthcare—A review. *Procedia Comput. Sci.* 2015;72:306–313. doi: 10.1016/j.procs.2015.12.145.

[8] Bray F., Ferlay J., Soerjomataram I. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J. Clin.* 2018;68:394–424. doi: 10.3322/caac.21492.

[9] Piñeros M., Znaor A., Mery L. A global cancer surveillance framework within noncommunicable disease surveillance: Making the case for population-based cancer registries. *Epidemiol. Rev.* 2017;39:161–169. doi: 10.1093/epirev/mxx003.

[10] Ma X., Yu H. Global burden of cancer. *Yale J. Biol. Med.* 2006;79:85–94.

[11] Agarap, A. F. M. (2018, February). On breast cancer detection: an application of machine learning algorithms on the wisconsin diagnostic dataset. In Proceedings of the 2nd international conference on machine learning and soft computing (pp. 5-9).

[12] V. Chaurasia, S. Pal, and B. Tiwari, "Prediction of benign and malignant breast cancer using data mining techniques," Journal of Algorithms & Computational Technology, vol. 12, no. 2, pp. 119-126, 2018.

[13] Fatima, N., Liu, L., Hong, S., & Ahmed, H. (2020). Prediction of breast cancer, comparative review of machine learning techniques, and their analysis. IEEE Access, 8, 150360-150376.

[14] Padhi, T., & Kumar, P. (2019, January). Breast Cancer Analysis Using WEKA. In 2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence) (pp. 229-232). IEEE.

[15] Thomas, T., Pradhan, N., & Dhaka, V. S. (2020, February). Comparative analysis to predict breast cancer using machine learning algorithms: a survey. In 2020 International Conference on Inventive Computation Technologies (ICICT) (pp. 192-196). IEEE.

[16] Shalini, M., & Radhika, S. (2020, February). Machine Learning techniques for Prediction from various Breast Cancer Datasets. In 2020 Sixth International Conference on Bio Signals, Images, and Instrumentation (ICBSII) (pp. 1-5). IEEE.

[17] Kabiraj, S., Raihan, M., Alvi, N., Afrin, M., Akter, L., Sohagi, S. A., & Podder, E. (2020, July). Breast cancer risk prediction using XGBoost and random forest algorithm. In 2020 11th international conference on computing, communication and networking technologies (ICCCNT) (pp. 1-4). IEEE.

[18] Marne, S., Churi, S., & Marne, M. (2020, March). Predicting breast cancer using effective classification with decision tree and k means clustering technique. In 2020 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 39-42). IEEE.

[19] Khourdifi, Y., & Bahaj, M. (2018, December). Applying best machine learning algorithms for breast cancer prediction and classification. In 2018 International conference on electronics, control, optimization and computer science (ICECOCS) (pp. 1-5). IEEE.

[20] Yarabarla, M. S., Ravi, L. K., & Sivasangari, A. (2019, April). Breast cancer prediction via machine learning. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI) (pp. 121-124). IEEE.

[21] Shen, L., Margolies, L.R., Rothstein, J.H. et al. Deep Learning to Improve Breast Cancer Detection on Screening Mammography. Sci Rep 9, 12495 (2019).

[22] Mori M, Akashi-Tanaka S, Suzuki S, Daniels MI, Watanabe C, Hirose M, Nakamura S. Diagnostic accuracy of contrast-enhanced spectral mammography in comparison to conventional full-field digital mammography in a population of women with dense breasts. Springer. 2016;24(1):104–10.

[23] Kurihara H, Shimizu C, Miyakita Y, Yoshida M, Hamada A, Kanayama Y, Tamura K. Molecular imaging using PET for breast cancer. Springer. 2015;23(1):24–32.

[24] Ch. Shravya, K. Pravalika, Shaik Subhani, "Prediction of Breast Cancer Using Supervised Machine Learning Techniques", International Journal of Innovative Technology and Exploring Engineering (IJITEE), Volume-8 Issue-6, April 2019.

[25] Min-Wei Huang,Chih-Wen Chen ,Wei-Chao Lin,Shih-Wen Ke,Chih-Fong Tsai, "SVM and SVM Ensembles in Breast Cancer Prediction", PLoS ONE 12(1): e0161501.

[26] Deepika Verma, Nidhi Mishra, "Comparative analysis of breast cancer and hypothyroid dataset using data mining classification techniques", 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)

[27] Wang Haifeng; Yoon Sang Won, "Breast Cancer Prediction Using Data Mining Method", IIE Annual Conference. Proceedings; Norcross (2015): 818-828

cancer prediction

ORIGINALITY REPORT

2%
SIMILARITY INDEX

PRIMARY SOURCES

1	www.researchgate.net Internet	19 words — 1%
2	Ismail ERBAS, David Alejandro VARGAS, Burak GUCLU. "FPGA Implementation of Multinomial Logistic Regression For Vibrotactile Feedback In a Robotic Hand", 2020 International Conference on e-Health and Bioengineering (EHB), 2020 Crossref	16 words — 1%
3	usir.salford.ac.uk Internet	12 words — < 1%
4	Yash Mate, Neelam Somai. "Hybrid Feature Selection and Bayesian Optimization with Machine Learning for Breast Cancer Prediction", 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), 2021 Crossref	8 words — < 1%
5	www.ncbi.nlm.nih.gov Internet	7 words — < 1%

EXCLUDE QUOTES	OFF	EXCLUDE SOURCES	OFF
EXCLUDE BIBLIOGRAPHY	ON	EXCLUDE MATCHES	OFF