PROSTATE CANCER DETECTION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS OF THE DEGREE OF

BACHELOR OF ENGINEERING

IN

INFORMATION TECHNOLOGY

BY

RAJ SHAH

MAHIPAL SUNDVESHA

MERLYN KOONAMPARAMPATH

UNDER THE GUIDANCE OF

PROF. MEENA UGALE

(Department of Information Technology)

INFORMATION TECHNOLOGY DEPARTMENT

XAVIER INSTITUTE OF ENGINEERING

UNIVERSITY OF MUMBAI

2021 – 2022

XAVIER INSTITUTE OF ENGINEERING

MAHIM CAUSEWAY, MAHIM, MUMBAI - 400016.

XAVIER INSTITUTE OF ENGINEERING

MAHIM CAUSEWAY, MAHIM,

MUMBAI - 400016

CERTIFICATE

This to certify that

RAJ JIGNESH SHAH	(53)
MAHIPAL SUNDVESHA	(61)
MERLYN KOONAMPARAMPATH	(27)

Have satisfactorily carried out the PROJECT work titled "**PROSTATE CANCER DETECTION"** in partial fulfillment of the degree of Bachelor of Engineering as laid down by the University of Mumbai during the academic year 2021-2022

Supervisor/Guide

Prof. Meena Ugale — **Head of Department**

Dr. Y.D Venkatesh — **Principal**

PROJECT REPORT APPROVAL FOR B.E.

This project report entitled "Prostate Cancer Detection"

By

RAJ JIGNESH SHAH	**(53)**
MAHIPAL SUNDVESHA	**(61)**
MERLYN KOONAMPARAMPATH	**(27)**

is approved for the degree of BACHELOR OF ENGINEERING.

Examiners

1. _____

2. _____

Supervisors

1. _____

2. _____

Date:

Place: MAHIM, MUMBAI

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' Ideas or words have been included; I have adequately cited and referenced the original sources.

I also declare that I have adhered to all the principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission.

I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which thus have not been properly cited or from whom proper permission have not been taken when needed.

RAJ JIGNESH SHAH (53) ----------

MAHIPAL SUNDVESHA (61) ----------

MERLYN KOONAMPARAMPATH (27) ----------

Date:

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Prostate Cancer is said to be the second leading cause of death in men. It is said to be a very slow growing cancer, showing no signs of it until the advanced stage. Different researches on AI algorithms processing different medical images like CT, MRI, Ultrasound have been massive over a few years. Using AI for prostate cancer management would make an impact in healthcare. With over 1.3M new cases every year worldwide, specialists working in this field would get a better understanding and can make more accurate predictions in cancer detection. The system will be built using Machine Learning algorithms for classification. Various image processing techniques would be applied to the dataset. The output obtained can be used to predicting whether the tumor is cancerous or not. Different algorithms like Support Vector Machine, Decision Tree, Random Forest, Gaussian Naïve Bayes, XGBoost, K-Means are considered for the system. The dataset consists of 100 records which included 38 benign cases and 62 malignant cases. Six different models, Decision Tree, Random Forest, K-Means, SVM, G. Naive Bayes, XGBoost are considered. The accuracy obtained from SVM and Random Forest is 95% which was the highest, and K-Means had the least accuracy at 67%.

Acknowledgement

We would like to thank Fr. (Dr). John Rose S.J. (Director of XIE) for providing us with suchan environment so as to achieve goals of our project and supporting us constantly.

We express our sincere gratitude to our Honourable Principal Dr. Y.D.Venkatesh for encouragement and facilities provided to us.

We would like to place on record our deep sense of gratitude to Prof Meena Ugale, Head of Dept. Of Information Technology, Xavier Institute of Engineering, Mahim, Mumbai, for her generous guidance help and useful suggestions.

With deep sense of gratitude we acknowledge the guidance of our project guide Prof Meena Ugale. The time-to-time assistance and encouragement by her has played an important rolein the development of our project.

We would also like to thank our entire Information Technology staff who have willingly co-operated with us in resolving our queries and providing us all the required facilities on time.

RAJ JIGNESH SHAH (43) ----------

MAHIPAL SUNDVESHA (61) ----------

MERLYN KOONAMPARAMPATH (27) ----------

Chapter 1

Introduction

1.1 Introduction

Prostate cancer is cancer that occurs in the prostate. The prostate is a small walnut-shaped gland in males that produces the seminal fluid that nourishes and transports sperm. Prostate cancer is one of the most common types of cancer. Many prostate cancers grow slowly and are confined to the prostate gland, where they may not cause serious harm. However, while some types of prostate cancer grow slowly and may need minimal or even no treatment, other types are aggressive and can spread quickly.

In prostate cancer, the use of AI overall has shown to be beneficial to aid in a standardized pathological grading to assess prostate cancer stratification and treatment. Additionally, AI shows promise in automating the assessment of characterization and severity of prostate cancer based on image-based tasks including in histopathologic, MRI, and biomarker diagnosis.

Furthermore, certain patients that are diagnosed with prostate cancer which is thought to be more indolent can continue with repeated forms of surveillance including prostate biopsies, PSA, and other forms of digital testing through MRI or rectal examinations unless they experience any physiological side effects. AI can help improve these forms of surveillance and will be amongst some of the essential tools to urological pathologists and to the field of urology as a whole as technology continues to improve and help patient prognosis over time.

1.2 Problem Definition

Computer-based decision-support systems and machine learning (ML) have the potential to revolutionize the healthcare system by playing advanced tasks that are presently appointed to specialists to enhance diagnostic accuracy, increase the potency of throughputs, improve clinical advancement, decrease human resource costs and improve treatment decisions. We aim to design a system using AI, which uses information derived from medical images that help to detect prostate cancer in men that can reduce discrepancies caused due to traditional tests like PSA, biopsy, and so on.

1.3 Scope of the Project

- The main aim is to detect the prostate cancer tumors present in the body.
- Reduction in dependency on PSA and Biopsy for detecting prostate cancer.
- To detect early stage cancer cells.

1.4 Existing System

Current screening on the basis of prostate-specific antigen (PSA) levels has a tendency towards both false positives and false negatives, both of which have negative consequences. Hence, in the existing system, a pipeline is developed in order to deal with imbalanced data and proposed algorithms to perform preprocessing on such datasets obtained from medical reports.[2]

In another system, the deep learning (DL) network is trained to extract and to learn its own features on the basis of the raw image to improve the classification of the image compared with the ML approach by using CNN, convolutional neural network.[3]

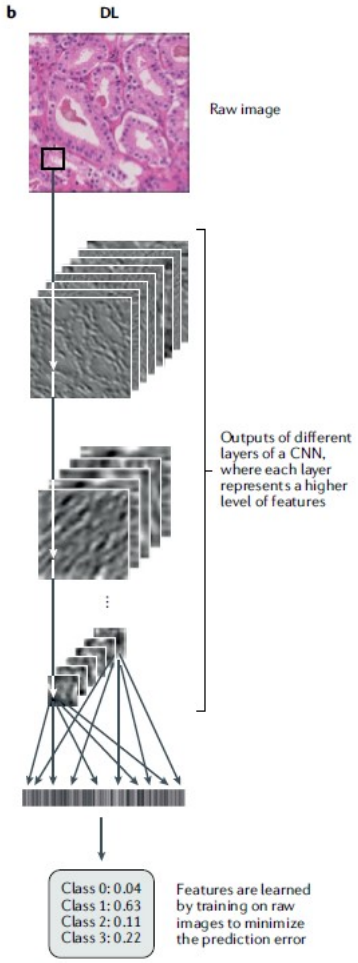


Figure 1.1: DL approach applied to a digital pathology image.[3]

1.5 Proposed System

This proposed system presents a comparison of machine learning (ML) algorithms: Support Vector Machine(SVM), Decision Tree(DT), Random Forest(RT), Gaussian Naïve Bayes(GNB), XGBoost(XGB), K-Means(KM). The algorithm that gives the best results will be supplied as a model to the website. The process of the proposed system is as follows :

- The home page consists of fields to input data related to various factors required to calculate results.
- The following page shows the result whether the cancer is malignant or benign.

Algorithms

Machine learning (ML) is an artificial intelligence (AI) technique that enables software to improve prediction accuracy without being particularly designed to do so. Using historical data as input, machine learning algorithms forecast new output values. The method through which an algorithm learns to increase its prediction accuracy is known as traditional machine learning. Supervised learning, unsupervised learning, semi-supervised learning, and reinforcement learning are the four basic methodologies.The algorithm to be used is determined by the sort of data they wish to predict.

Decision Trees are a type of Supervised Machine Learning in which data is continually separated based on a parameter. The tree can be explained using two entities: decision nodes and leaves. The decisions or final outcomes are represented by the leaves, and the data is split at the decision nodes.

K-Means Clustering is a type of Unsupervised Learning method that divides an unlabeled dataset into groups. It's an iterative technique that splits the unlabeled dataset into k clusters, with each dataset belonging to only one group with similar qualities. It allows us to divide data into multiple groups and gives a straightforward method for determining group categories in an unlabeled dataset without any training.

A common Supervised Learning technique for handling classification and regression problems is the Support Vector Machine (SVM). The goal of the SVM method is to determine the best line or decision boundary for categorising n-dimensional space into classes so that subsequent data points can be easily placed in the right category. The optimum choice boundary is known as a hyperplane. Support vectors, or extreme points or vectors, are chosen by SVM to aid in the formation of the hyperplane.

The Naive Bayes method is a supervised learning algorithm that uses the Bayes theorem to solve classification problems. It's widely used in text classification issues where a big training dataset is required. The Naive Bayes Classifier is a basic and effective classification method for developing fast machine learning models that can make quick predictions. It's a probabilistic classifier, meaning it makes predictions based on the probability of an object. The Gaussian model assumes that characteristics are distributed normally. The model implies that continuous values are drawn from a Gaussian distribution if predictors take continuous values rather than discrete values.

Random forest is a supervised machine learning approach for solving classification and regression issues. It uses the majority vote for classification and the average for regression to generate decision trees from various samples. The Random Forest Algorithm's ability to handle data sets with both continuous and categorical variables, as in regression and classification, is one of its most important features. It outperforms the competition when it comes to classification problems.

XGBoost is a high-speed and high-performance gradient boosted decision tree implementation. Gradient boosting, multiple additive regression trees, stochastic gradient boosting, and gradient boosting machines are all terms used to describe this approach. Boosting is an ensemble strategy that involves adding new models to old models to remedy faults. Models are added in a logical order until no further enhancements are needed. Gradient boosting is a technique that involves creating new models that forecast the residuals or errors of previous models, which are then combined to form the final prediction. When adding new models,

it employs a gradient descent approach to minimise loss.

Dataset
The dataset finalized for the project is 'Prostate Cancer Dataset' from Kaggle website.[6]
The total dataset size is 4.4 KB and the no. of observations are 100.

Chapter 2

Review of Literature

Sr No	Year	Authors	Title	Description
1	2021	Chalida Aphinives, Potchavit Aphinives	Artificial intelligence development for detecting prostate cancer in MRI	Detection of Prostate Cancer lesion in T2W MRI using ML algorithms.
2	2021	Derek J Van Booven, Manish Kuchakulla, Raghav Pai, Fabio S Frech, Reshna Ramasahayam, Pritika Reddy, Madhumita Parmar, Ranjith Ramasamy, Himanshu Arora	A Systematic Review of Artificial Intelligence in Prostate Cancer	Cancer Detection system by using ANNs based DL network.
3	2021	Octavian Sabin Tătaru, Mihai Dorin Vartolomei, Jens J. Rassweiler, Osan Virgil, Giuseppe Lucarelli, Francesco Porpiglia, Daniele Amparore, Matteo Manfredi, Giuseppe Carrieri, Ugo Falagario, Daniela Terracciano, Ottavio de Cobelli, Gian Maria Busetto, Francesco Del Giudice, Matteo Ferro	Artificial Intelligence and Machine Learning in Prostate Cancer Patient Management—Current Trends and Future Perspectives	Using ANN algorithm like DCNNs for diagonis and prognoses of Prostate Cancer on image datasets.
4	2020	Liron Pantanowitz, Gabriela M Quiroga-Garza, Lilach Bien, Ronen Heled, Daphna Laifenfeld, Chaim Linhart, Judith Sandbank, Anat Albrecht Shach, Varda Shalev, Manuela Vecsler, Pamela Michelow, Scott Hazelhurst, Rajiv Dhir	An artificial intelligence algorithm for prostate cancer diagnosis in whole slide images of core needle biopsies: a blinded clinical validation and deployment study	Using images from the needle biopsies test, the CNN algorithm is used to detect Cancer.
5	2019	S. Larry Goldenberg, Guy Nir, Septimiu E. Salcudean	A new era: artificial intelligence and machine learning in prostate cancer	Prostate Segmentation using CNN based DL networks on MRI and Ultrasound images.
6	2019	Henry Barlow, Shunqi Mao, Matloob Khushi	Predicting High-Risk Prostate Cancer Using Machine Learning Methods	Determining High Risk Cancer using ML algorithms like SVM on different factors like rate of change of PSA, Age, BMI, and Race.
7	2018	Islam Reda MSc, Ashraf Khalil PhD, Mohammed Elmogy PhD, Ahmed Abou El-Fetouh PhD, Ahmed Shalaby PhD, Mohamed Abou El-Ghar PhD, Adel Elmaghraby PhD, Mohammed Ghazal PhD, Ayman El-Baz PhD	Deep Learning Role in Early Diagnosis of Prostate Cancer	A computer-aided diagnostic system was developed for early diagnosis of Prostate Cancer by using ML algorithms such as KNN.

Table 2.1: Literature Review Summary Table.

1. Chalida Aphinives, Potchavit Aphinives "Artificial intelligence development for detecting prostate cancer in MRI", Egyptian Journal of Radiology and Nuclear Medicine, 2021. [1]

The paper states that AI for the detection of Prostate Cancer lesion in T2W MRI can be easily developed. It can predict one third of PCa lesions correctly after training with only 160 images and 'Custom Vision' service by using ML algorithms. However, the AI development is further required as DL algorithms will need additional programming, and the result should be interpreted along with radiologist.

2. Derek J Van Booven, Manish Kuchakulla, Raghav Pai, Fabio S Frech, Reshna Ramasahayam, Pritika Reddy, Madhumita Parmar, Ranjith Ramasamy, Himanshu Arora, "A Systematic Review of Artificial Intelligence in Prostate Cancer", Dove Press journal: Research and Reports in Urology, 2021. [8]

This study has tried to detect cancer by using ANNs. The neural network in this study was trained on conventional factors like age, stage, bone scan findings, grade, PSA and treatment along with 2 experimental markers of immunostaining for bcl-2 and p53. By including experimental markers for testing into building the networks, each piece of data could impact the model accuracy profoundly. AI ANNs can allow the development of effective classification system for prostate cancer risk stratification.

3. Octavian Sabin Tătaru, Mihai Dorin Vartolomei, Jens J. Rassweiler, Osan Virgil, Giuseppe Lucarelli, Francesco Porpiglia, Daniele Amparore, Matteo Manfredi, Giuseppe Carrieri, Ugo Falagario, Daniela Terracciano, Ottavio de Cobelli, Gian Maria Busetto, Francesco Del Giudice, Matteo Ferro, Artificial Intelligence and Machine Learning in Prostate Cancer Patient Management—Current Trends and Future Perspectives, MPDI, 20 February 2021. [7]

The paper states about different approaches in implementing AI algorithms in Digital Pathology, Diagnostic Imaging, Genomics and Treatment of Prostate Cancer. Deep learning methods seem to be the most appropriate models to be applied in histopathology, especially through image data sets analysis and classification. In Prostate Cancer, ANN algorithms especially DCNNs are promising for diagnosis and playing a predictive role in the prognoses of the disease.

4. Liron Pantanowitz, Gabriela M Quiroga-Garza, Lilach Bien, Ronen Heled, Daphna Laifenfeld, Chaim Linhart, Judith Sandbank, Anat Albrecht Shach, Varda Shalev, Manuela Vecsler, Pamela Michelow, Scott Hazelhurst, Rajiv Dhir, , "An artificial intelligence algorithm for prostate cancer diagnosis in whole slide images of core needle biopsies: a blinded clinical validation and deployment study", Lancet Digital Health Vol. 2, August 2020. [4]

This study reports the successful development, external clinical validation, and deployment in clinical practice of an AI-based algorithm to accurately detect, grade, and evaluate clinically relevant findings in digitised slides of prostate CNBs.

5. S. Larry Goldenberg, Guy Nir, Septimiu E. Salcudean, "A new era: artificial intelligence and machine learning in prostate cancer", Nature Reviews Urology, 15 May 2019.[3]

In this paper, for MRI and ultrasonic images, Prostate Segmentation is done using DL. A CNN-based DL network could be used to enable accurate detection of low to medium risk cancer in Ultrasound. For MRI images, an Imaged-Based tecnique developed by ML along with funtional and texture feautures included, combining CNN to train the datasets.

6. Henry Barlow, Shunqi Mao, Matloob Khushi, "Predicting High-Risk Prostate Cancer Using Machine Learning Methods", MDPI, 2 September 2019. [2]

The system aimed to evaluated the efficiency of ML methods for Prostate Cancer. Different factors such as rate of change of PSA, age, BMI, and Race were analysed by using SVM. Including the rate of change of PSA and age increased accuracy whereas BMI and race had a minimal effect.

7. Islam Reda MSc , Ashraf Khalil PhD , Mohammed Elmogy PhD , Ahmed Abou El-Fetouh PhD , Ahmed Shalaby PhD , Mohamed Abou El-Ghar PhD , Adel Elmaghraby PhD , Mohammed Ghazal PhD , Ayman El-Baz PhD, "Deep Learning Role in Early Diagnosis of Prostate Cancer", SAGE Journals, Technology in Cancer Research and Treatment, 27 May 2018. [5]

A computer-aided diagnostic system was developed for early diagnosis of Prostate Cancer by using ML algorithms such as KNN.

Chapter 3

Proposed System

Description of Dataset
For the Prostate Cancer data set from kaggle, a total of 100 cases with ten attributes were collected. The characteristic "diagnosis" is characterised as the quantifiable, with zero indicating that patients do not have prostate cancer and one indicating that they have. The data set included 38 cases of prostate cancer that were not diagnosed and 62 cases of prostate cancer that were diagnosed. The data set is made up of 100 observations and ten variables (eight numeric variables and one categorical variable, as well as ID) which are as follows: Id, Radius, Texture, Perimeter, Area, Smoothness, Compactness, Diagnosis Result, Symmetry, Fractal dimension. [6]

id	diagnosis_result	radius	texture	perimeter	area	smoothness	compactness	symmetry	fractal_dimension
1	M	23	12	151	954	0.143	0.278	0.242	0.079
2	B	9	13	133	1326	0.143	0.079	0.181	0.057
3	M	21	27	130	1203	0.125	0.16	0.207	0.06
4	M	14	16	78	386	0.07	0.284	0.26	0.097
5	M	9	19	135	1297	0.141	0.133	0.181	0.059
6	B	25	25	83	477	0.128	0.17	0.209	0.076
7	M	16	26	120	1040	0.095	0.109	0.179	0.057
8	M	15	18	90	578	0.119	0.165	0.22	0.075
9	M	19	24	88	520	0.127	0.193	0.235	0.074
10	M	25	11	84	476	0.119	0.24	0.203	0.082
11	M	24	21	103	798	0.082	0.067	0.153	0.057
12	M	17	15	104	781	0.097	0.129	0.184	0.061
13	B	14	15	132	1123	0.097	0.246	0.24	0.078
14	M	12	22	104	783	0.084	0.1	0.185	0.053
15	M	12	13	94	578	0.113	0.229	0.207	0.077
16	M	22	19	97	659	0.114	0.16	0.23	0.071
17	M	10	16	95	685	0.099	0.072	0.159	0.059
18	M	15	14	108	799	0.117	0.202	0.216	0.074
19	M	20	14	130	1260	0.098	0.103	0.158	0.054
20	B	17	11	87	566	0.098	0.081	0.189	0.058

Figure 3.1: Images of Prostate Cancer Dataset[6]

3.1 Block Diagram

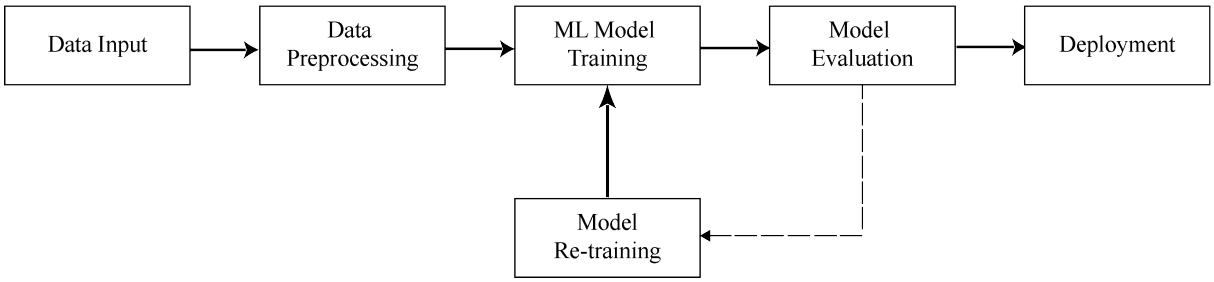


Figure 3.2: Block Diagram

The elements of the system are represented in the block diagram. The data is fed as input into the system. The data is then prepared to be used for the ML model. One of the most significant techniques for data preparation is data pre-processing. Data Augmentation is also done at this step. The next step is to train our machine learning model. Various ML algorithms are used to extract the features from data, and make predictions. The model is then evaluated with the help of various statistical methods. It is not obligatory that the first model be the best. The first model serves as a baseline model, which can be trained again to improve accuracy. Finally, we must deploy the model. This step involves applying and migrating the model to operations for their use.

3.2 Analysis and Design

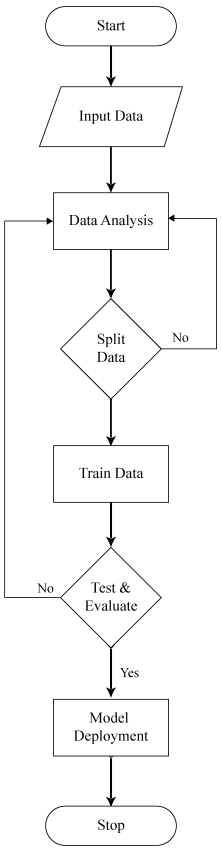


Figure 3.3: Flowchart Diagram

Algorithm:

Step 1: Start.

Step 2: Input Data.

Step 3: Analysing the data for missing values, relations between the variables, etc.

Step 4: Splitting the data into training and testing sets.

Step 5: Training data using different Machine Learning models

Step 6: Testing the model performance and evaluating the same. If accurate results are obtained, go to step7 else go to step 3.

Step 7: Model is deployed for operation.

Step 7: End.

Chapter 4

Implementation Strategy

4.1 Software Requirements

- Computer with minimum 4GB RAM.
- Python along with Flask and other libraries as and when required.
- Machine Learning Libraries.
- Anaconda Navigator
- VSCode Editor

4.2 Methodology

The dataset considered has only 100 observations. During the preprocessing stage, the data was augmented to 500 records. The process of modifying, or augmenting a dataset with extra data is known as data augmentation. This supplementary input can be anything from visuals to text, and it aids the performance of machine learning algorithms. Many AI systems require data augmentation since accuracy rises with the amount of training data. Basic data augmentation can considerably enhance accuracy in classification and segmentation. Data collection and labelling can be time-consuming and costly for machine learning models. By modifying datasets, data augmentation methods can be utilised to lower operating costs. Since the dataset considered consisted of only 100 records, augmentation was done on the data. The number of records was increased copying the complete dataset five times which resulted in 500 records.

Label encoding is the process of translating labels into a numeric format so that they can be read by machines. Under the column 'Diagnosis Result' the values Benign (b) and Malignant (M) were converted to 0 and 1 respectively by using Label Encoder. Many machine learning estimators require dataset standardisation: if the individual features do not more or less resemble standard normally distributed data, they may perform poorly. The task of standardisation is carried out by StandardScaler. StandardScaler's concept is that it will turn your data into distribution with a mean of 0 and a standard deviation of 1. In this dataset, since each variable has values in different scales, Standard Scaler is used to standardise the data to have a common scale while building the machine learning model.

Different models were developed using various algorithms and their accuracy and classification report was produced. We mostly deal with two types of problems in machine learning: classification and regression. For these purposes, there are various distinct types of algorithms. However, we must choose an algorithm that performs well on the given data. When it comes to classification, ensemble approaches such as Random Forest, Decision Tree, and XGBoost algorithms have shown excellent results. These algorithms provide excellent accuracy while operating at a high rate.

Extreme Gradient Boosting Algorithm, or XGBoost, is an ensemble method that operates by boosting trees. Gradient Boosting is the name given to XGboost since it uses a gradient descent technique. The whole point is to fix the model's prior error, learn from it, and enhance performance in the next step. The

prior results have been corrected, and performance has been improved.

This process is repeated until there is no more room for improvement. For this form of predictive algorithm, regularisation is the most important aspect. It is simple to use and provides good accuracy. Because of its ability to tolerate missing values and avoid overfitting, this technique is widely utilised. A booster, learning rate, objective, and other hyperparameters are used in this type of algorithm once again. Among all the models developed, XGBoost was selected as the best model because of its characteristics. The model was deployed for Website operation.

Flask is a Python-based web application framework. The Web Application Framework, or simply Web Framework, is a collection of tools and modules that allow web application developers to build apps without having to deal about low-level issues like protocols and thread management. Flask is built on the foundations of the Werkzeug WSGI toolkit and the Jinja2 template engine. The Web Server Gateway Interface (WSGI) is a standard for creating a uniform interface between an online server and web applications. Werkzeug is a WSGI toolkit that allows you to create requests, response objects, and other useful tasks. This makes it possible to construct a web framework on top of it. Werkzeug is one of the foundations of the Flask framework. Jinja2 is a popular Python templating engine. To render dynamic web pages, a web templating system combines a template with a specific data source. Flask is also known as a micro framework. Its goal is to keep an application's core basic but expandable. Flask lacks a built-in abstraction layer for database management, as well as functionality for form validation. Instead, Flask allows you to use extensions to add this functionality to your programme.

The Website consists of a homepage which includes input fields where one can enter its respective data into it. After all the fields are filled, the predict button is to be clicked to calculate the result. This leads to the Output page. If the values entered are for Benign cases, then the Benign page shows up. And as for Malignant cases, if the values entered are for the same, the Malignant page shows up.

4.3 Implementation

CODE:

Without Data augmentation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
data= pd.read_csv('D:Prostate_Cancer.csv')
X=data
y=data
print(X)
print(y)
print(X.isna().sum())

#Histograms
ig = plt.figure(figsize=(20, 30))
plt.suptitle('Histograms of Numerical Columns', fontsize=20)
for i in range(1, data.shape[1] + 1):
plt.subplot(8, 3, i)
f = plt.gca()
f.axes.get_yaxis().set_visible(False)
```

```
f.set_title(data.columns.values[i - 1])
vals = np.size(data.iloc[:, i - 1].unique())
plt.hist(data.iloc[:, i - 1], bins=vals, color='Blue')

#HeatMap
plt.figure(figsize=(15, 15), dpi=80)
plt.title('Heat Map')
sn.heatmap(X, annot=True)

plot = sn.countplot('diagnosis_result',hue='radius', data=data)
plt.ylabel('Total')
plt.show()

X=data.drop(data.columns[[0,1]], axis = 1)
y=data['diagnosis_result']
print(X)
print(y)
print(X.isna().any())
print(y.isna().any())

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
print(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
print(X_train)

#DecisionTree
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 2)
classifier.fit(X_train, y_train)

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, recall_score
y_pred = classifier.predict(X_test)
cm =confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#KMeans
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters = 2, init = 'k-means++', random_state = 3)
kmeans.fit(X_train, y_train)
```

```
y_kmeans = kmeans.predict(X_test)
cm =confusion_matrix(y_test, y_kmeans) print(accuracy_score(y_test, y_kmeans))
print(classification_report(y_test, y_kmeans))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#RandomForest
from sklearn.ensemble import RandomForestClassifier
randomforest= RandomForestClassifier(random_state= 42, verbose=1)
randomforest.fit(X_train,y_train)
random_pred = randomforest.predict(X_test)
cm= confusion_matrix(y_test, random_pred)
print(accuracy_score(y_test, random_pred))
print(classification_report(y_test, random_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#SVM
from sklearn.svm import SVC
classifier = SVC(kernel='linear', random_state=42)
classifier.fit(X_train, y_train)
y_pred= classifier.predict(X_test)
cm= confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#GNaiveBayes
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
cm= confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#XGBoost
from xgboost import XGBClassifier
classifier = XGBClassifier(use_label_encoder=False, eval_metric= 'error')
```

```
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
cm= confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")
```

With Data augmentation

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sn
data= pd.read_csv('D:Prostate_Cancer.csv')
X=data
y=data
print(X)
print(y)
print(X.isna().sum())

import pandas as pd
X_aug = pd.DataFrame(X)
data = pd.concat([X_aug]*5, ignore_index=True)
print(data)

X=data.drop(data.columns[[0,1]], axis = 1)
y=data['diagnosis_result']
print(X)
print(y)
print(X.isna().any())
print(y.isna().any())

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
print(y)

from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
print(X_train)

#DecisionTree
from sklearn.tree import DecisionTreeClassifier
classifier = DecisionTreeClassifier(criterion = 'entropy', random_state = 2)
```

```
classifier.fit(X_train, y_train)

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, recall_score
y_pred = classifier.predict(X_test)
cm =confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#KMeans
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters = 2, init = 'k-means++', random_state = 3)
kmeans.fit(X_train, y_train)
y_kmeans = kmeans.predict(X_test)
cm =confusion_matrix(y_test, y_kmeans) print(accuracy_score(y_test, y_kmeans))
print(classification_report(y_test, y_kmeans))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#RandomForest
from sklearn.ensemble import RandomForestClassifier
randomforest= RandomForestClassifier(random_state= 42, verbose=1)
randomforest.fit(X_train,y_train)
random_pred = randomforest.predict(X_test)
cm= confusion_matrix(y_test, random_pred)
print(accuracy_score(y_test, random_pred))
print(classification_report(y_test, random_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#SVM
from sklearn.svm import SVC
classifier = SVC(kernel='linear', random_state=42)
classifier.fit(X_train, y_train)
y_pred= classifier.predict(X_test)
cm= confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")
```

```
#GNaiveBayes
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(X_train, y_train)
y_pred = gnb.predict(X_test)
cm= confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test, y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")

#XGBoost
from xgboost import XGBClassifier
classifier = XGBClassifier(use_label_encoder=False, eval_metric= 'error')
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
cm= confusion_matrix(y_test, y_pred)
print(accuracy_score(y_test,y_pred))
print(classification_report(y_test, y_pred))

ax= sn.heatmap(cm, annot=True, cmap="Blues", xticklabels=['Benign', 'Malignant'],
yticklabels=['Benign','Malignant'])
ax.set_xlabel("Predicted Labels")
ax.set_ylabel("True Labels")
```

Website

Prostate_detection_model.py

```
from xml.etree.ElementPath import xpath_tokenizer
import pandas as pd
import numpy as np
import pickle
import pandas as pd

data= pd.read_csv('Prostate_Cancer.csv')

X=data
y=data

X_aug = pd.DataFrame(X)
data = pd.concat([X_aug]*5, ignore_index=True)

X=data.drop(data.columns[[0,1]], axis = 1)
y=data['diagnosis_result']

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(y)
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20)

from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
sc = scaler.fit(X_train)
X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
std = np.sqrt(sc.var_)
np.save('std.npy',std )
np.save('mean.npy',sc.mean_)

from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, recall_score
from xgboost import XGBClassifier
classifier = XGBClassifier(use_label_encoder=False, eval_metric= 'error')
classifier.fit(X_train, y_train)
y_pred= classifier.predict(X_test)
print(y_pred)
print(accuracy_score(y_test,y_pred))
print(recall_score(y_test, y_pred))

filename = 'finalized_model.sav'
pickle.dump(classifier,open(filename,'wb'))
```

app.py

```
import numpy as np
from flask import Flask, request, jsonify, render_template
import pickle

app = Flask(__name__)
loaded_model = pickle.load(open('finalized_model.sav','rb'))

@app.route('/')
def home():
return render_template('index.html')

@app.route('/predict',methods=['POST'])
def predict():

s = np.load('std.npy')
m = np.load('mean.npy')
int_features = [float(x) for x in request.form.values()]
final_features = (np.array([int_features]-m))/s
predicted = loaded_model.predict(final_features)
return render_template('output.html', data=predicted)

if __name__ == "__main__":
app.run(debug=True)
```

OUTPUT:

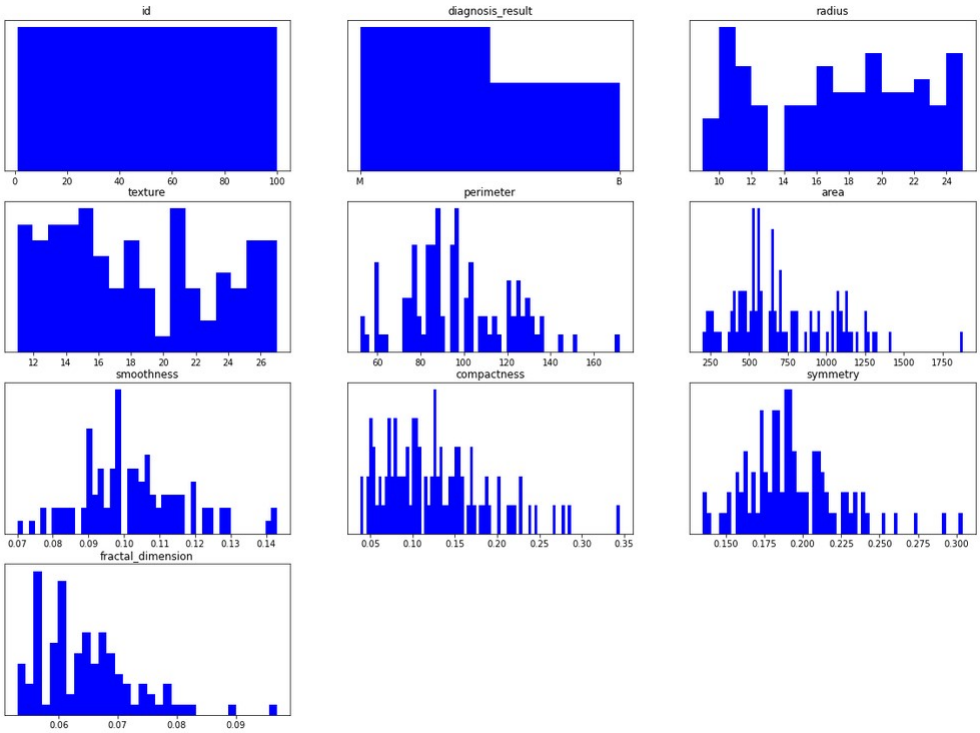


Figure 4.1: Histograms of Numerical Columns

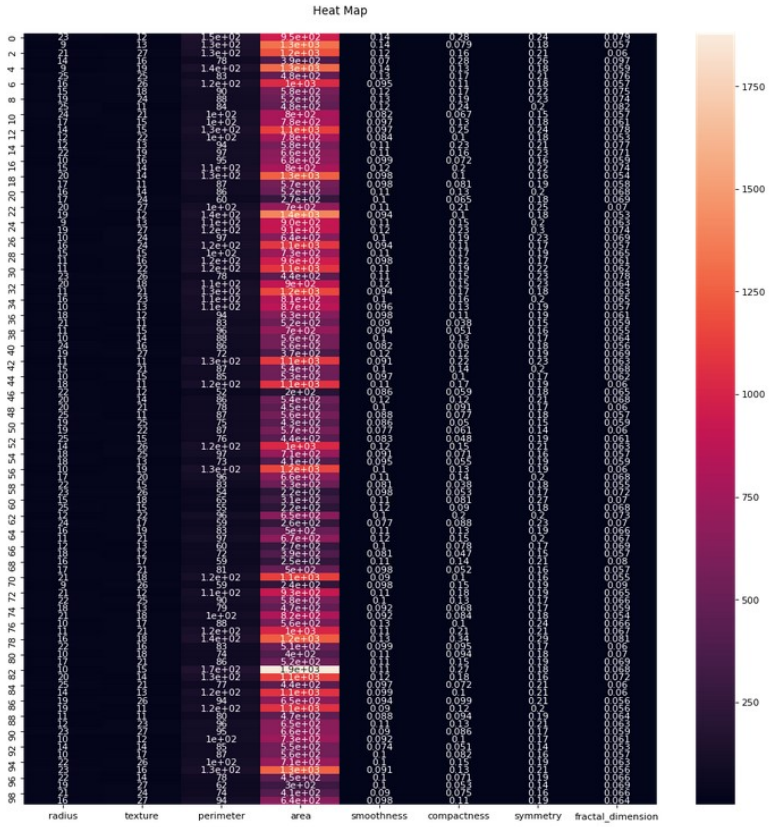


Figure 4.2: HeatMap

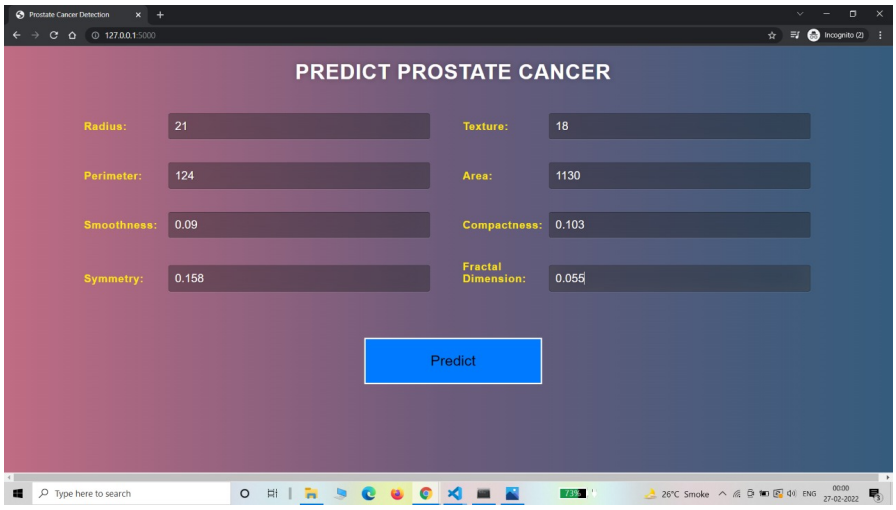


Figure 4.3: Website Homepage

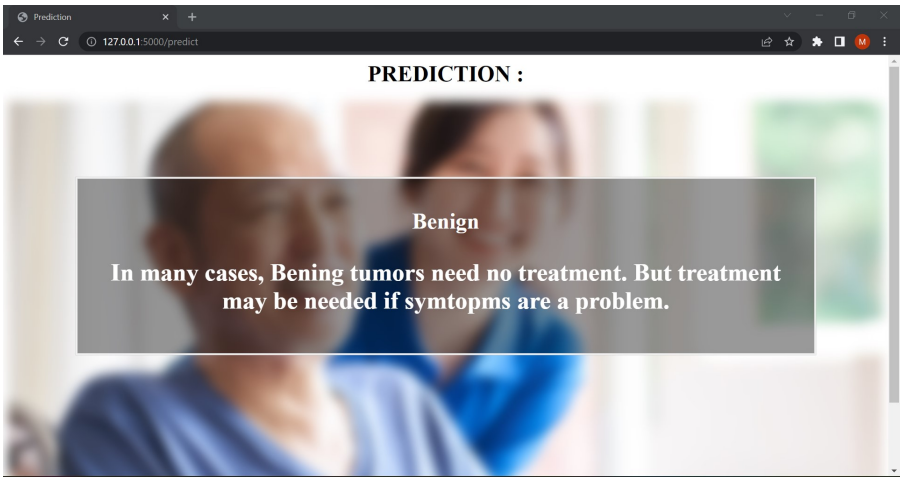


Figure 4.4: Output: Benign

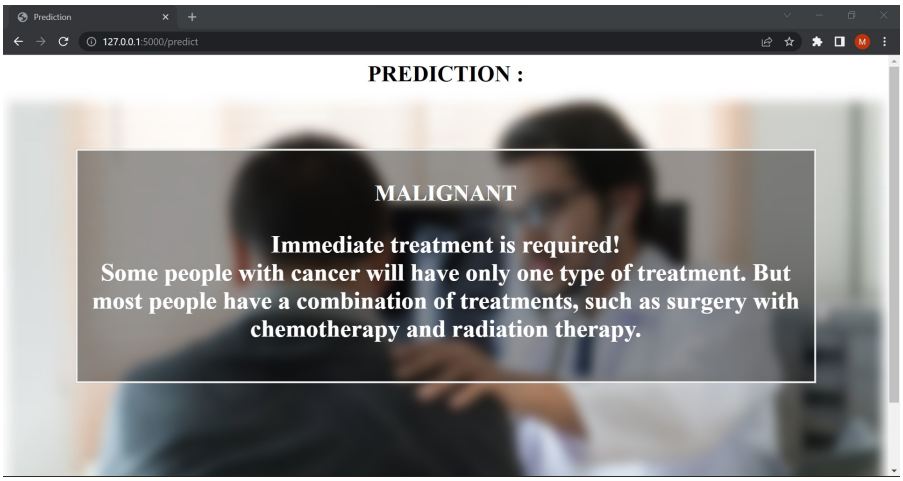


Figure 4.5: Output: Malignant

Chapter 5

Results

5.1 Confusion Matrix

The images below show the Confusion Matrix (CM) of the models. Overall accuracy, sensitivity, specificity, and precision are calculated using the CM data. The percentage of correctly categorised test samples is known as accuracy. The percentage of tuples that are positive and predicted to be positive by the classifier is called sensitivity. Specificity is the inverse of sensitivity; it refers to actual negatives predicted as negatives by the classifier. Precision is also known as exactness and is synonymous with sensitivity. CM of models without Data Augmentation are depicted on the left side where as the CM of models with Data Augmentation are depicted on the right.

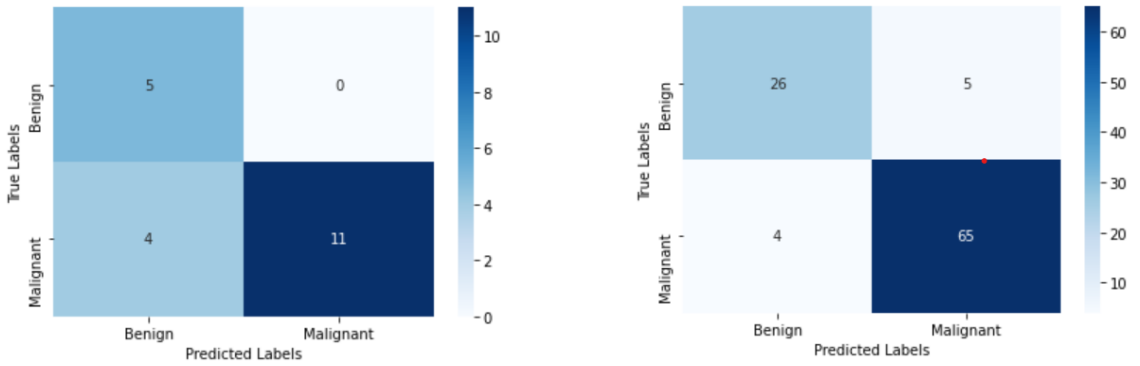


Figure 5.1: Confusion Matrix of Decision Tree

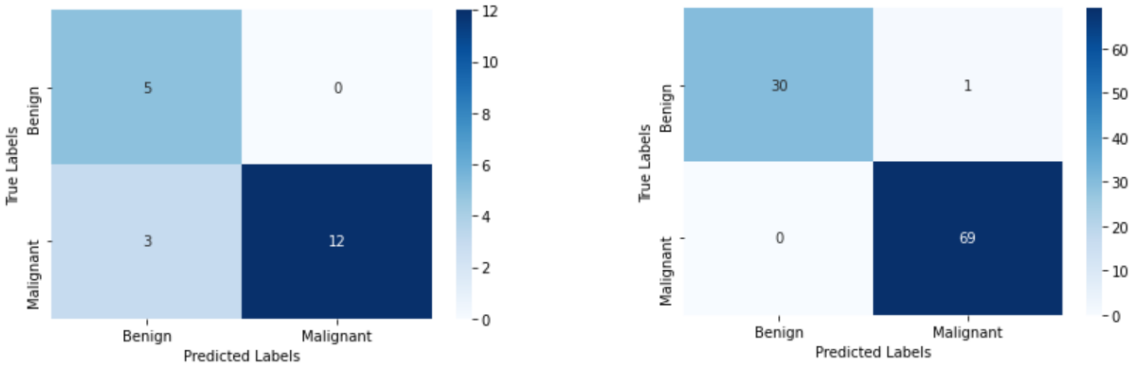
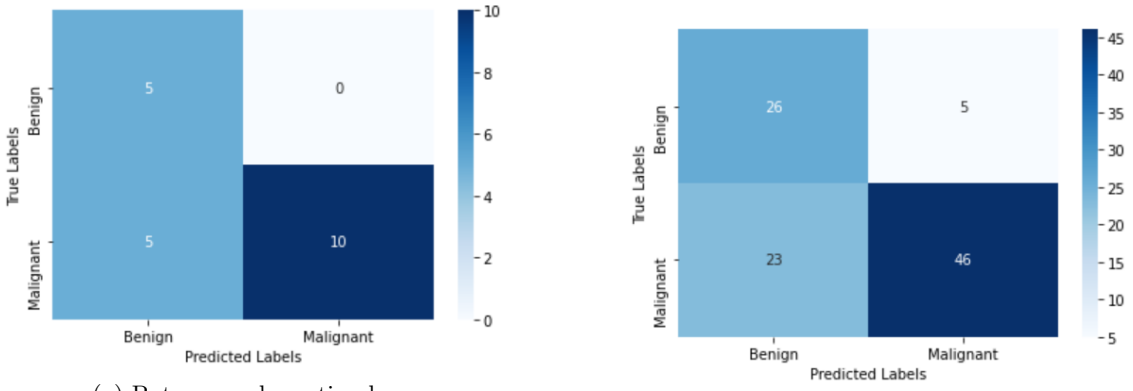


Figure 5.2: Confusion Matrix of Random Forest



(a) Put your sub-caption here

Figure 5.3: Confusion Matrix of G. Naive Bayes

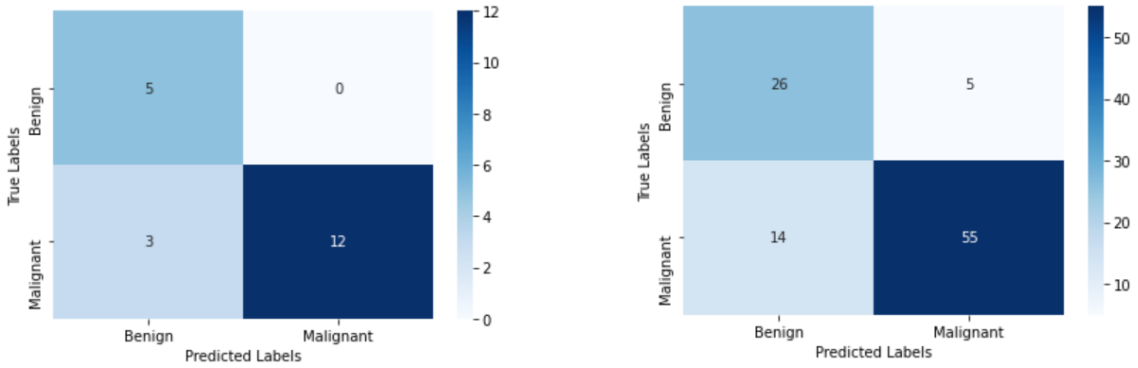


Figure 5.4: Confusion Matrix of SVM

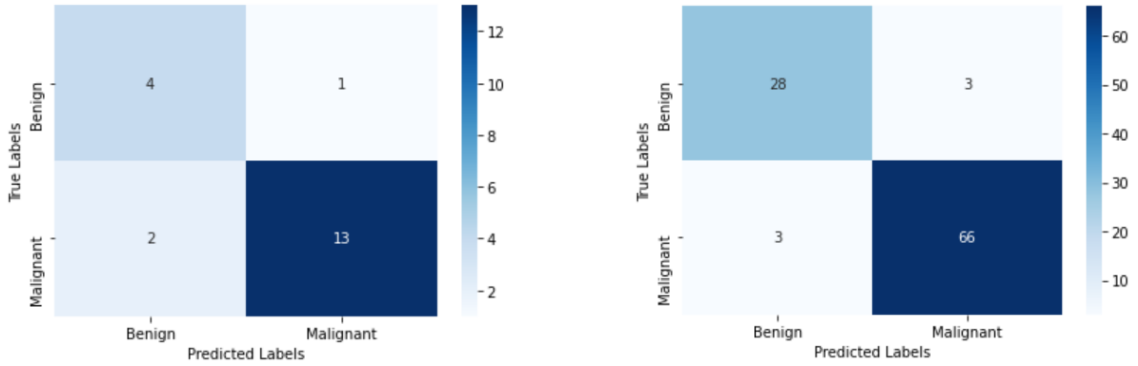


Figure 5.5: Confusion Matrix of XGBoost

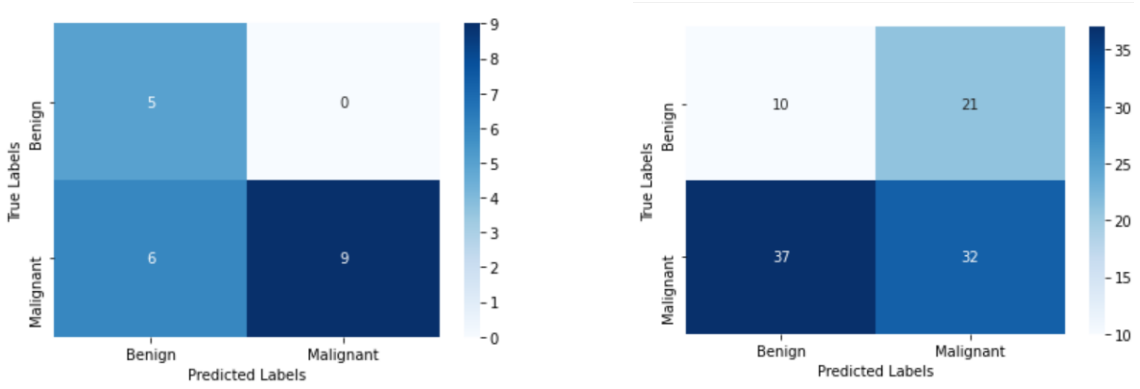


Figure 5.6: Confusion Matrix of K-Means

5.2 Accuracy

The result analysis (Table 5.1) shows that without Data Augmentation, Gaussian Naive Bayes and SVM provided the best result i.e, 85% whereas K-Means has the least result of 25%. With Data Augmentation, SVM and Random Forest provided the best result with an accuracy of 95% whereas K-Means had the least result of 67%. Fig 5.7 shows the performance comparison of the models.

Algorithm	Result without Data Augmentation	Result with Data Augmentation
Decision Tree	80	91
Random Forest	80	95
G. Naive Bayes	85	83
SVM	85	95
XGBoost	75	93
K-Means	25	67

Table 5.1: Summary of Results.

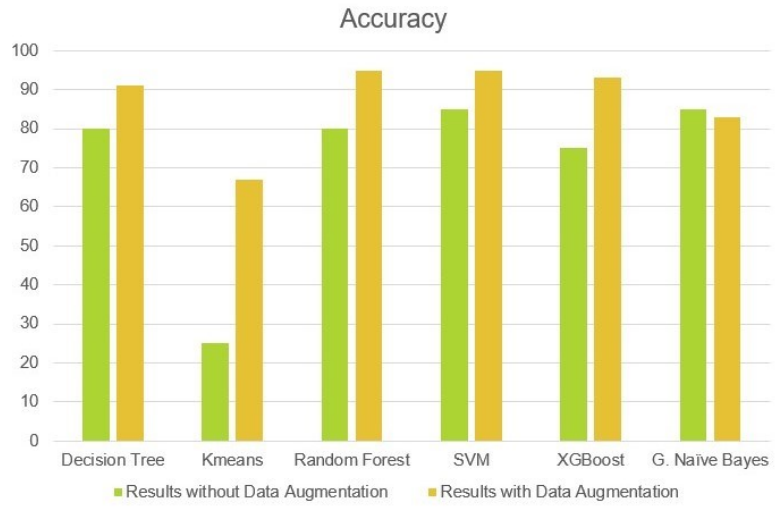


Figure 5.7: Graphical representation of Various Algorithms

Chapter 6

Conclusion

Certain patients that are diagnosed with prostate cancer which is thought to be more indolent can continue with repeated forms of surveillance including prostate biopsies, PSA, and other forms of digital testing through MRI or rectal examinations unless they experience any physiological side effects. AI can help improve these forms of surveillance and will be amongst some of the essential tools to urological pathologists and to the field of urology as a whole as technology continues to improve and help patient prognosis.

The task done in Semester 6 was Literature Survey in order to learn more about this field. Different datasets were analysed and the most favourable one was selected. Techniques involved in classification of cancer were studied and Machine Learning was found to be suitable for our requirements. Different algorithms like Support Vector Machine, Decision Tree, Random Forest, Gaussian Naïve Bayes, XGBoost, K-Means are considered for the system.

Last Semester we started with the implementation of our project. We are done with Data cleaning and processing of our Dataset. We also began with our model development which is to be completed in the next semester along with fine tuning of the model. A Review paper was completed on the summary of state-of-the-art CNN-based approaches applied for Prostate Cancer Identification which was also successfully published.

The fine tuning of our model was done this semester. The best model was selected for Deployment. Website was also developed using Flask. A Technical Paper is also completed based on our project.

Chapter 7

References

[1] Chalida Aphinives and Potchavit Aphinives. "Artificial intelligence development for detecting prostate cancer in MRI". In: *Egyptian Journal of Radiology and Nuclear Medicine* 52.1 (Mar. 2021), p. 87. ISSN: 2090-4762. DOI: 10.1186/s43055-021-00467-4. URL: https://doi.org/10.1186/s43055-021-00467-4.

[2] Henry Barlow, Shunqi Mao, and Matloob Khushi. "Predicting High-Risk Prostate Cancer Using Machine Learning Methods". In: *Data* 4.3 (2019). ISSN: 2306-5729. DOI: 10.3390/data4030129. URL: https://www.mdpi.com/2306-5729/4/3/129.

[3] S. Larry Goldenberg, Guy Nir, and Septimiu E. Salcudean. "A new era: artificial intelligence and machine learning in prostate cancer". In: *Nature Reviews Urology* 16.7 (July 2019), pp. 391–403. ISSN: 1759-4820. DOI: 10.1038/s41585-019-0193-3. URL: https://doi.org/10.1038/s41585-019-0193-3.

[4] Liron Pantanowitz et al. "An artificial intelligence algorithm for prostate cancer diagnosis in whole slide images of core needle biopsies: a blinded clinical validation and deployment study". In: *The Lancet Digital Health* 2.8 (2020), e407–e416. ISSN: 2589-7500. DOI: https://doi.org/10.1016/S2589-7500(20)30159-X. URL: https://www.sciencedirect.com/science/article/pii/S258975002030159X.

[5] I. Reda et al. "Deep Learning Role in Early Diagnosis of Prostate Cancer". In: *Technol Cancer Res Treat* 17 (Jan. 2018), p. 1533034618775530.

[6] Sajid Saifi. *Prostate cancer*. Oct. 2018. URL: https://www.kaggle.com/datasets/sajidsaifi/prostate-cancer.

[7] Octavian Sabin Tătaru et al. "Artificial Intelligence and Machine Learning in Prostate Cancer Patient Management—Current Trends and Future Perspectives". In: *Diagnostics* 11.2 (2021). ISSN: 2075-4418. DOI: 10.3390/diagnostics11020354. URL: https://www.mdpi.com/2075-4418/11/2/354.

[8] D. J. Van Booven et al. "A Systematic Review of Artificial Intelligence in Prostate Cancer". In: *Res Rep Urol* 13 (2021), pp. 31–39.

7.1 Publications

Koonamparampath, M., Shah, R., Sundvesha, M. and Ugale, M., "A Review on Prostate Cancer Detection using CNN." In: International Journal for Research in Applied Science and Engineering Technology, (2022), 10(3), pp.948-953. doi: 10.22214/ijraset.2022.40747.