#### ABSTRACT

This study is to predict lung cancer using several Machine Learning algorithms and lifestyle characteristics. The LUNA16 Dataset is collected from the Kaggle repository to test classification results of the Random Forest, Support Vector Machines, Logistic Regression and KNN algorithms. Among the classification metrics were properly classified occurrences, F-Measure, Precision, Accuracy, and Recall. Random Forest out performed the other algorithms for big datasets with the same number of features, according to the findings.The other algorithms were less precise than Random Forest, which attained 98%% accuracy. This research is significant because precise cancer categorization and early identification can significantly reduce death rates. The trained model would forecast the level of the lung cancer, such as low,medium, or high and it would even predict the type of lung cancer that could occur.

#### TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **No** | **TITLE** | **PAGE No** |
|  | ABSTRACT | iv |
|  | LIST OF FIGURES | vi |
|  | LIST OF ABBREVATIONS | vii |
| 1 | INTRODUCTION | 1 |
| 2 | LITERATURE SURVEY | 3 |
| 3 | PROBLEM STATEMENT | 8 |
| 4 | OBJECTIVE | 9 |
| 5 | PROPOSED BLOCK DIAGRAM | 10 |
|  | 5.1 PREPROCESSING | 10 |
|  | 5.2 CLASSIFICATION | 11 |
| 6 | TOOLS USED | 15 |
|  | 6.1 GOOGLE COLAB | 15 |
|  | 6.2 PYCHARM | 15 |
| 7 | SOFTWARE DESCRIPTION | 16 |
| 8 | RESULTS AND DISCUSSION | 17 |
| 9 | CONCLUSION | 20 |
| 10 | FUTURE SCOPE | 21 |
| 11 | REFERRENCES | 22 |
| 12 | APPENDIX   1. SOURCE CODE 2. SCREENSHOT 3. COURSE COMPLETION CERTIFICATES 4. CONTEST PARTICIPATION 5. PLAGIARISM REPORT   **v** | 24 |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE**  **No** | **TITLE** | **PAGE No** |
| 5.1 | PROPOSED SYSTEM DIAGRAM | 08 |
| 5.2.1 | WORKING OF RANDOM FOREST | 10 |
| 5.2.2 | WORKING OF LOGISTIC REGRESSION | 10 |
| 5.2.3 | WORKING OF SVM | 11 |
| 5.2.4 | WORKING OF KNN | 12 |
| 8.1 | SAMPLE RESULT | 14 |
| 8.2 | KNN RESULT | 16 |
| 8.3 | RANDOM FOREST RESULT | 16 |
| 8.4 | LOGISTIC REGRESSION RESULT | 16 |
| 8.5 | SVM RESULT | 16 |

**LIST OF ABBREVIATIONS**

**HTML – Hyper Text MarkUp Language**

**CSS-Cascading Style Sheets**

**RNN-Recurrent Neural Networks**

**CNN-Convolutional Neural Networks**

**SVM-Support Vector Machine**

### CHAPTER – 1 INTRODUCTION

Machine Learning is the field of study that gives computers the capability to learn without being explicitly programmed. ML is one of the most exciting technologies that one would have ever come across. As it is evident from the name, it gives the computer that makes it more similar to humans: The ability to learn. Machine learning is actively being used today, perhaps in many more places than one would expect.

Machine learning, as a powerful approach to achieve Artificial Intelligence, has been widely used in pattern recognition, a very basic skill for humans but a challenge for machines. Nowadays, with the development of computer technology, pattern recognition has become an essential and important technique in the field of Artificial Intelligence. The pattern recognition can identify letters, images, voice or other objects and also can identify status, extent or other abstractions.

Since the computer was invented, it has begun to affect our daily life. It improves the quality of our lives; it makes our life more convenient and more efficient. A fascinating idea is to let a computer think and learn as a human. Basically, machine learning is to let a computer develop learning skills by itself with given knowledge. Pattern recognition can be treated like computer being able to recognize different species of objects. Therefore, machine learning has close connection with pattern recognition.

Machine Learning is a scientific research of statistical procedures and methods which they are used by computer systems designed to perform such functions without specific instructions, rather than trusting in the models and conclusions. This is believed to be part of an artificial intelligence. Machine Learning algorithms sets up a mathematical model based on data examples called "training data" to make predictions without the completion of a task being explicitly programmed.

CNN(Convolutional neural network) algorithm is a popular machine learning algorithm that has found applications in various fields such as computer vision, text classification, and medical diagnosis. In this research, CNN algorithm and its use in predicting cancer with lifestyle factors. The Decision Tree is a well-known machine learning technique that may also be utilized for classification and regression applications. Decision Trees are built by recursively splitting the input space into subsets depending on one or more input feature values. As a consequence, a tree-like structure is formed, with each leaf node representing a categorization label.

The benefit of CNN algorithm includes:

* Overcoming the issue of overfitting.
* Categorical, and binary data, making it suitable for high and Less sensitivity to aberrant data in training data.
* Setting parameters is simple, so there is no need to prune the plants.
* Automatically generated variable accuracy and significance

This paper concentrate on the performance of the classification of CNN for the prediction. The objective of this comparison is creating a baseline, which will be useful for the classification scenarios of prediction of the cancer. It will also help in the selection of appropriate model. The remaining portions of the paper are structured as follows: Classification methods, such as the Random Forest, KNN, SVM, Logistic Regression are described. The CNN method is a well-known machine learning approach that has been used effectively in a many of applications, including medical diagnostics. The CNN algorithm is employed in this study to predict cancer based on lifestyle characteristics.

### CHAPTER – 2 LITERATURE SURVEY

##### A Survey on Prediction of benign and malignant breast Cancer using Data Mining Techniques by Vikas Chaurasia, Saurabh pal, BB Tiwari: [1]

This publication stated that Breast cancer is the second most common malignancy in women after cervical cancer. In 2004, around 1.1 million instances were reported. The observed rates of this malignancy grow with industrialization and urbanization, as well as with early detection services. It is still considerably morewidespread in high-income nations, but it is rapidly spreading to middle- and low-income countries, including parts of Africa, Asia, and Latin America. Breast cancer kills less than half of all patients and is the main cause of cancer mortality in women, accounting for 16% of all cancer fatalities globally. The goal of the study work is to offer a report on breast cancer in which used existing technical breakthroughs to construct survivorshipprediction models. utilizing a large dataset (683 breast cancer cases), developed prediction models utilizingthree common data mining methods (Nave Bayes, RBF Network, and J48). For performance comparison, it is also employed 10-fold cross-validation methods to calculate the unbiased estimate of the three prediction models. The results (based on average accuracy Breast Cancer dataset) showed that the Nave Bayes is the best predictor with 97.36% accuracy on the holdout sample (this prediction accuracy is better than any reported in the literature), RBF Network came in second with 96.77% accuracy, and RBF Network came in third with 96.77% accuracy.

**A Review on Early Diagnosis of Breast Cancer Prediction using Random Forest Classifiers Anisha P R, Kishor Kumar Reddy C, K Apporva and Meghana Mangipudi C:** [2]

This study is that Breast cancer is one of the most terrifying diseases and a leading cause of mortality in women. Late discovery of Breast Cancer may substantially diminish survival chances; as a remedy, the automatic diseasedetection system assists the medical sector in diagnosing and analysing, providing fast reaction, dependability,efficacy, and lowering the danger of death. In this study, we show how a Machine Learning Technique called Random Forest Classifier may be used to detect breast cancer. This classifier divides the data into multiple treesand determines whether a person is at risk of developing breast cancer or not. The accuracy of this model is 98%.

##### A Review on Breast Cancer prediction Using Fine Needle Aspiration Features and Upsampling with Supervised Machine Learning by Rahman Sahfique : [3]

Breast cancer is the second largest cause of mortality in women from cancer. Early identification of breast cancercan save lives, but the typical method requires a battery of laboratory tests performed by medical professionals.To eliminate human error and accelerate breast cancer detection, an autonomous system that can execute the diagnostic reliably and quickly is necessary. Despite research efforts for automated cancer detection systems, there is a significant gap between the required and supplied accuracy of existing techniques.

To address this issue, this study presents a method for predicting breast cancer by choosing the best fine needleaspiration characteristics. Several feature selection approaches, including as principal component analysis, singular vector decomposition, and chi-square (Chi2), are used to improve prediction accuracy. Extensive studies are carried out with various features and feature set sizes in order to determine the ideal feature set. Theeffect of unbalanced and balanced data is also explored utilizing the SMOTE technique. To improve classification accuracy, six classifiers are tuned: random forest, support vector machine, gradient boosting machine, logistic regression, multilayer perceptron, and K-nearest neighbors (KNN). On the given dataset, theresults show that KNN beats all other classifiers.

##### A Survey on Machine Learning for prediction of in – hospital mortality in lung cancer patients admittedto intensive care unit by Tianzhi Huang, Dejin Le, Lili Yuan, Shoujia Xu, Xiulan peng: [4]

This study was to use machine learning algorithms to predict in-hospital mortality in critically unwell lung cancer patients in the intensive care unit (ICU). The data for the training cohort came from the Medical Information Mart for Intensive Care-IV (MIMIC-IV), while the data for the validation cohort came from the Medical Information Mart for ICU Collaborative Research Database (MICU-CRD). Several machine learning methods, such as logistic regression, random forest, decision tree, LightGBM, XGBoost, and an ensemble model integrating random forest, LightGBM, and XGBoost, were used. Each model's performance was assessed usingcriteria such as AUC (area under the receiver operating curve), accuracy, F1 score, and recall. Shapley Additive explanations (SHAP) values were also calculated to determine the relevance of each feature in the prediction.

##### A Survey on Polygenic Risk scores and breast cancer risk Prediction by Eleanor Roberts, Sacha Howell, DGareth Evans : [5]

This study reviews Polygenic Risk Scores (PRS) are an important component of accurate breast cancer risk prediction and have the potential to improve screening and preventative measures. PRS integrate the risk from single nucleotide polymorphisms (SNPs) related with breast cancer in Genome Wide Association Studies (GWAS) and explain more than 30% of breast cancer heritability. When included into risk models, the more personalised risk assessment obtained by PRS helps identify women at higher risk of developing breast cancer and facilitates the implementation of stratified screening and preventative interventions. This review discusses the importance of PRS in breast cancer risk prediction, as well as the development of PRS and their clinical use.It is also investigated the significance of PRS within more established risk prediction models that include recognized traditional risk factors, and it discussed how PRS interacts with these variables and their ability to predict breast cancer subtypes. Several problems must be overcome before PRS may be applied on a population-wide basis. The use of PRS among women of non-White European background, where PRS has been demonstrated to have diminished risk prediction both in discrimination and calibration, is perhaps the most pressing of these. It address the latest developments in the development and use of PRS in non- white European communities. PRS are a tremendous advancement in breast cancer risk prediction, and their further developmentwill surely improve customization.

##### A Survey on Random Forest for Breast Cancer Prediction by T L Octaviani and Z Rustam: [6]

This article offers the prediction of one of the leading causes of mortality in the United States. Breast cancer fatalities in Indonesia reached 21,287 in 2017, accounting for 1.27% of total deaths, according to WHO data. Delays in discovering the state of breast cancer in women with breast cancer result in greater mortality, poor prognosis, and shorter survival rates, all of which are connected with lesser breast cancer awareness and recommended non-adherence to screening. It is presented a random forest for breast cancer prediction in this study.Random forest is a huge data classification algorithm that is one of several classification algorithms. Random forest classification is used on cancer microarray data to improve classification accuracy and reliability.

5

##### A Survey on Diagnosis of Breast Cancer using Random Forest by Manas Minnor, Veeky Baths: [7]

This study presents that Predicting Cancer at present times approaches need professional consulting, which is costly and time-consuming, and so may not be available to many. The goal of this study is to train and test supervised machine learning models for the accurate and efficient diagnosis of breast cancer. The Wisconsin Breast Cancer Database dataset describes 30 characteristics of cell nuclei, including their radius, texture, and concavity. There are 569 occurrences in all, 212 of which are malignant tumors. Because the Random Forest method beats other algorithms in categorizing breast tumors as malignant or benign, it was chosen as our primary model. It is trained on two separate subsets of the dataset, each containing 16 and 8 features found usingseveral feature selection approaches. The Random Forest models are evaluated on a holdout set after hyperparameter adjustment and achieve accuracies of 100% and 99.30%, respectively. The models are also compared to four different machine learning classification algorithms: SVM, Decision Tree, Multilayer Perceptron, and K-Nearest Neighbors. The findings show that Random Forest is the best strategy for detectingbreast cancer.

##### A Review on Machine Learning Algorithms for Breast Cancer prediction and Diagnosis by Mohammed Amine Naji, Sanaa El Filali, Kawtar Aarika, EL Habib Benlahmari, Rachida Ait Abekohahid, Oliver Debauche: [8]

This Paper cited that the number of fatalities from breast cancer is growing dramatically each year. It is the mostcommon kind of cancer and the leading cause of death in women globally. Any advancement in the prediction and detection of cancer sickness is critical to living a healthy life. As a result, high accuracy in cancer prognosisis essential for updating therapy aspects and patient survival standards. Machine learning techniques have become a research hotspot and have been proven to be a strong approach in the process of prediction and earlydiagnosis of breast cancer. In this work, it used five machine learning algorithms on the Breast Cancer Wisconsin dataset: Support Vector Machine (SVM), Random Forest, Logistic Regression, Decision Tree (C4.5),and K-Nearest Neighbours (KNN).

6

##### A Survey on Machine Learning Techniques for Breast Cancer Prediction by Varsha Nemade, Vishal Fegade:

Breast cancer is the leading cause of death among women. Breast cancer prediction is a difficult job in medical data analysis. Doctors and pathologists needed automated tools to make decisions and distinguish between malignant and benign tumours. A machine learning (ML) algorithm may assist a lot in making judgements andperforming diagnoses based on data acquired by the medical industry. Several studies demonstrate that ML approaches can aid with decision making in breast cancer prediction. On a breast cancer dataset, It uses variousML Classification techniques: Nave Bayes (NB), Logistic regression (LR), Support vector machine (SVM), K-Nearest Neighbour (KNN), Decision Tree (DT), and ensemble techniques: Random forest (RF), Adaboost, and XGBoost and evaluated their performance using various performance measures. It was discovered that both thedecision tree and the XGBoost classifier had the greatest accuracy 97% across all models, with the highest AUC0.999 achieved for the XGBoost classifier.

### CHAPTER – 3 PROBLEM STATEMENT

Traditional lung cancer detection technologies are essential, they do have limits that can restrict their efficiency. Inaccessibility and the high cost of diagnostic and screening testing are two important restrictions. Cancer Diagnostic and Screening Tests Are Inaccessible: Due to a variety of causes, many people have difficulty getting cancer detection services. Geographic constraints might make it difficult for people living in distant or rural locations to access cancer screening and diagnostic services. In certain areas, limited infrastructure, transit challenges, or a lack of healthcare services can all contribute to this inaccessibility.

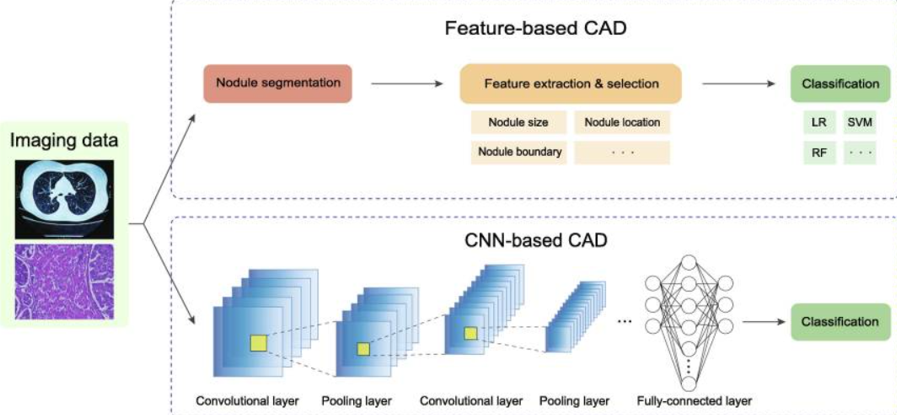
Furthermore, people with low-income or underserved neighborhoods may experience socioeconomic challenges, such as a lack of health insurance or limited financial means, that prohibit them from seeking and getting cancer screening services. Cancer Diagnostic and Screening procedures Have High expenditures: Cancer diagnostic and screening procedures frequently include complex technology, laboratory studies, and medical skills, which can result in large expenditures. These costs include the cost of the tests as well as charges for laboratory processing, radiological services, and consultations with healthcare specialists.

These high prices may put cancer screening programs out of reach for many people, particularly those without appropriate health insurance or with low financial resources. As a result, those who cannot afford these tests may postpone or skip cancer screenings, resulting in delayed diagnosis and potentially inferior treatment outcomes.

### CHAPTER – 4 OBJECTIVE OF THE PROJECT

* + 1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
    2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
    3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

### CHAPTER – 5 PROPOSED BLOCK DIAGRAM

****

**Figure 5.1 Proposed system diagram**

### Pre-processing:

Image has been collected from LIDC-LDRI. The original image was full of noise and for that first we have applied histogram equalization on the image to enhance the image and then on the equalized image median filter has been applied to remove the noise which was already present in the image after getting the noise free image we have applied some more noise in the image yield more clearer picture then again noise has been removed using median filter. Generally median filter is non linear digital filtering technique and it is also used as smoothing of images as it don’t blur the edges completely as compare to other filtration technique like Gaussian filter or average filter.

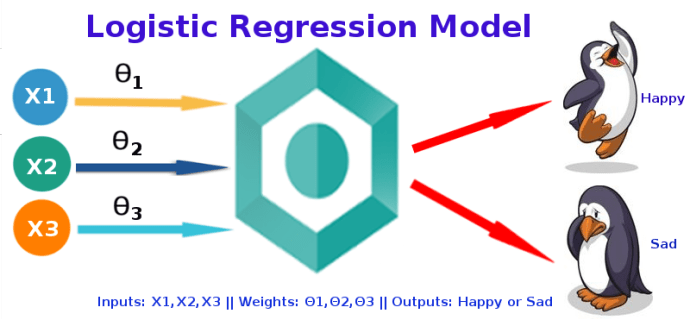
### Segmentation:

Image segmentation is a method of partitioning the image into various parts. After pre- processing the image on the pre-processed image segmentation is applied to acquire the information from the image. For image segmentation first we have applied edge detection technique through edge detection we can segment the boundary of the image for edge detection prewitt operator has been used, on that operator threshold has been applied so that after edge detection the intensity value which is less than threshold is removed and the intensity value which is higher than or equal to threshold will consider for further segmentation after getting the segmented image by edge detection we will apply watershed segmentation on the output image. Watershed segmentation takes the concept topographical landscape with ridge and valley which is defined by a gray level with respective pixel or gradient magnitude. There exist various ways to segment using watershed segmentation here we have used watershed segmentation using gradient. The gradient magnitude is used to preprocess the gray scale image; it has high pixel value along the object edge and low pixel value in another left region. And through this we can get the final segmented image through which we can extract features.

* 1. **Feature Extraction Layer**

The output generated by segmentation is used for feature extraction. By doing feature extraction we have extracted two types of feature one is region based another is texture based region based we have extracted feature like area in context to image means pixel of the image, perimeter in context image mean vector containing the distance around the boundary of each region in the image, centroid means the centre of mass of the region and it is in 1 X 2 vector form, image and based on texture we have extracted feature like mean is used to find average intensity, standard deviation is used to measure average contrast, smoothness used to measure relative smoothness of the intensity in the region, entropy is used to measure randomness using statistical approach of texture based.

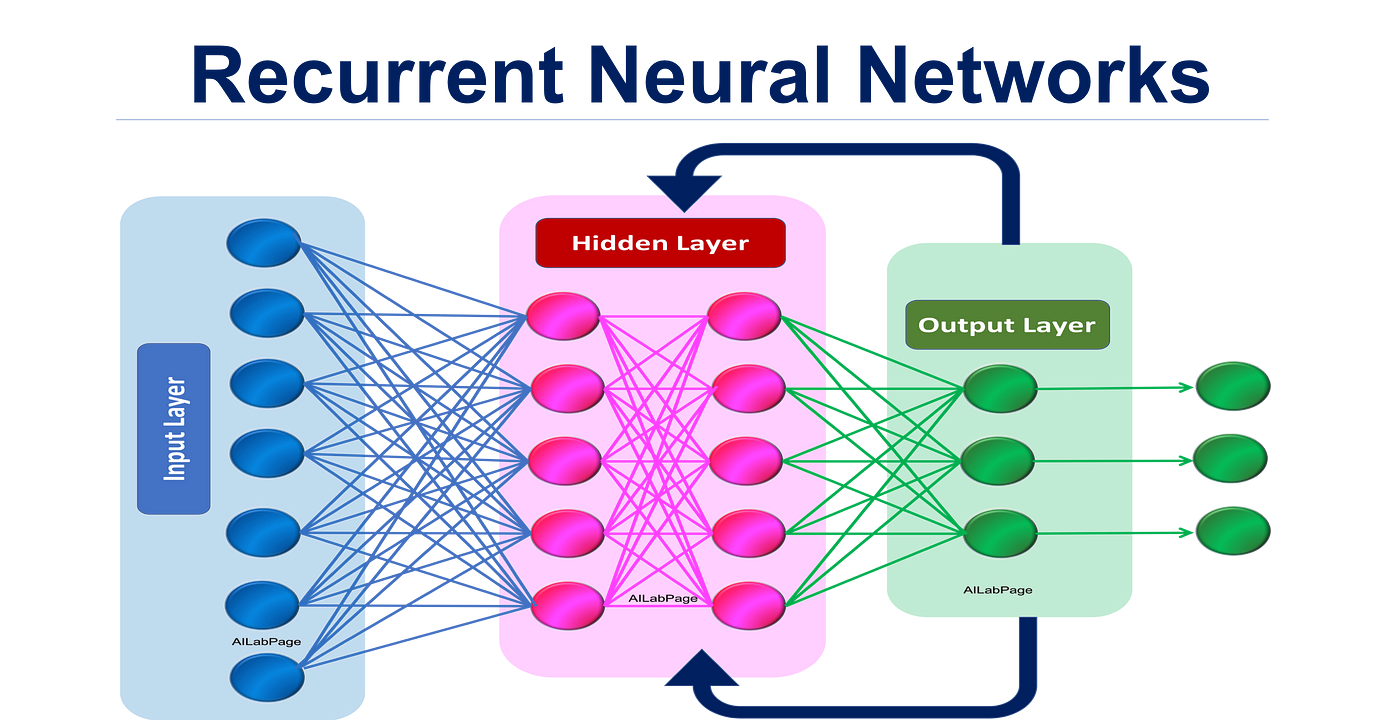
**5.4 Classification Layer**

After feature extraction we will apply classification technique on both the feature to compare at which feature extraction which machine learning algorithm is giving more accuracy. Machine learning algorithm which has been used is support vector machine, artificial neural network and Random forest. After applying classification technique, it can be predicted that the tumour is cancerous or not and at which feature we are getting more accurate prediction.

**Figure 5.2 Working of Logistic Regression**

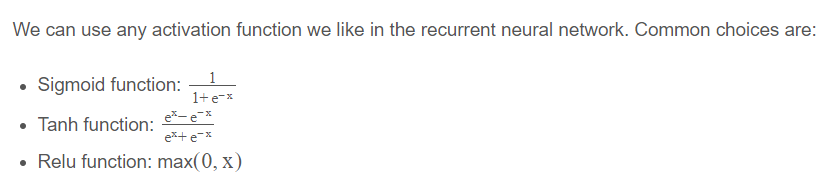
##### Mathematical Formula of Logistic Regression:

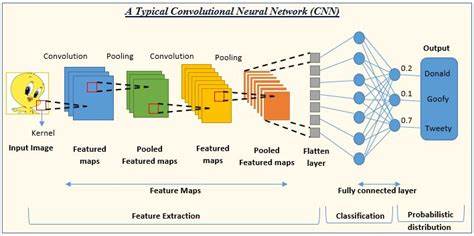
##### See the source image



**Figure 5.2: Working of RNN]**

##### Mathematical Formula of RNN:





**Figure 5.3: Working of CNN**

##### Mathematical Formula of CNN:

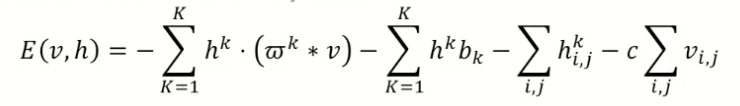


Figure [5.1], [5.2], [5.3] describes the working of the algorithms. Decision Trees are built from pre-classified data and focus on finding the best split features to classify the data,Random Forest is an ensemble learning method that builds multiple Decision Trees using randomized subsets of the data and features to improve the model's accuracy and robustness. Random Forest is an effective method for dealing with missing data, noisy data, and outliers.

The process of building multiple trees and aggregating their predictions helps to smooth out theeffects of noise and outliers, resulting in a more stable and accurate model.

### CHAPTER – 6 TOOLS USED

##### Google Colab

The cloud-based platform Google Colab is helpful for creating algorithms for spotting fake images. It gives users access to a GPU, which is crucial for deep learning model training, which calls for a lot of processing power. Additionally, Google Colab may be utilized for model assessment, deployment, and hyper parameter tweaking. The dataset may be preprocessed by downsizing the photographs, making them grayscale, and leveling the pixel values before training the model. The ideal combination of parameters may be determined by running many training sessions with various combinations of parameters using Google Colab. By calculating multiple evaluation metrics, the platform mayalso be used to assess how well the model performs on the test dataset. After the model has been trained, it maybe deployed by being converted into a production-ready format and being uploaded to a cloud-based server or edge device.

##### PyCharm

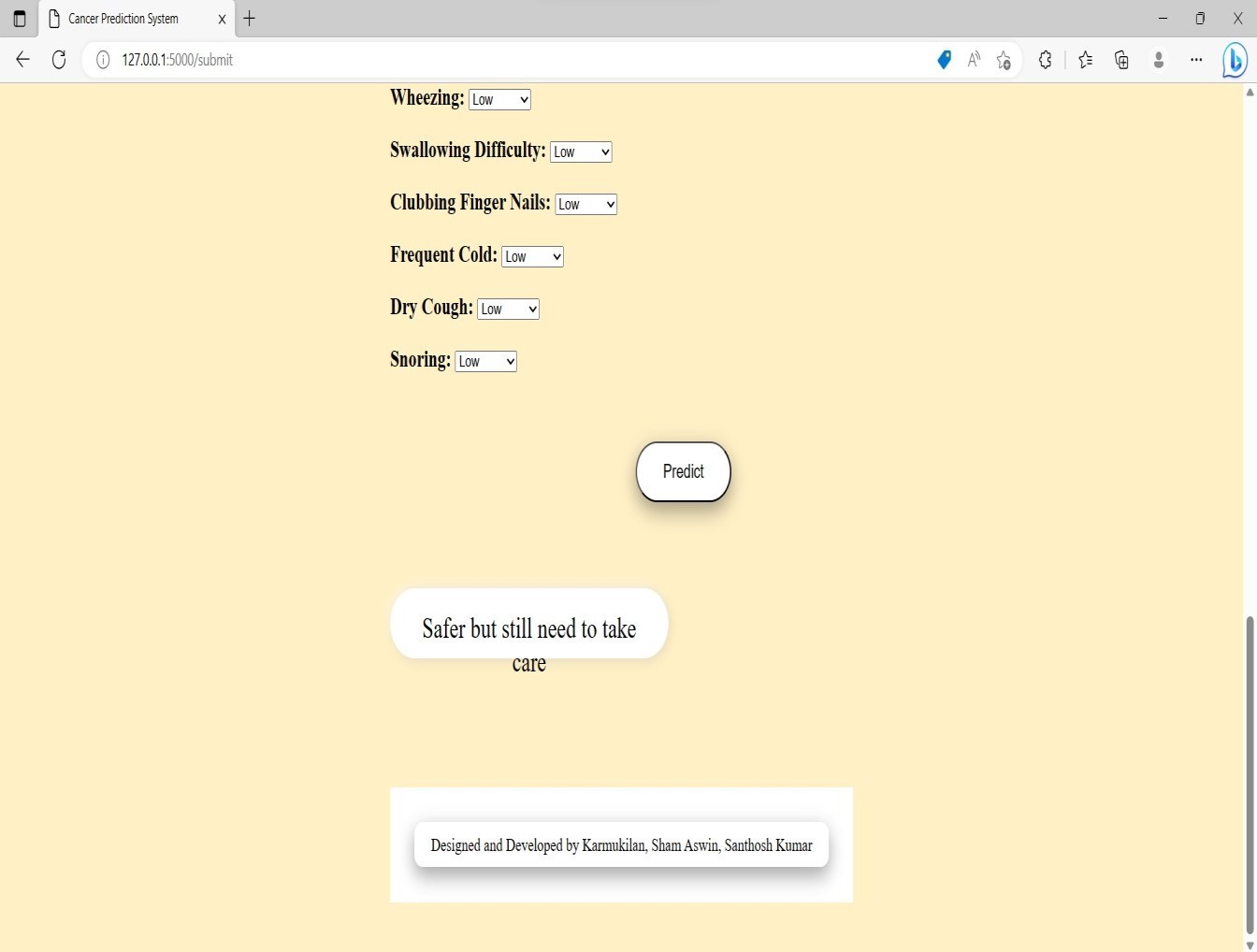
PyCharm has features like as code completion, debugging tools, an integrated terminal, and version control integration that can helps with Flask development. It also supports Flask-specific plugins and extensions, which can improve your development experience even more. PyCharm provides intelligent code completion, syntax highlighting, and error checking to help developers create code more effectively and with fewer errors. Integrated debugger: PyCharm's debugger makes it simple to debug Flask apps. To discover and resolve errors, developers can use breakpoints, walk through code, and check variables. Flask-specific templates: PyCharm offers Flask-specific project templates. These templates give the essential directory structure, configuration files,and example code as a starting point. PyCharm works smoothly with the Flask development server, allowing developers to launch and test Flask apps straight from the IDE. PyCharm has support for common version control systems like as Git, making it easy to maintain Flask projects and interact with team members. Databasesupport: PyCharm has database tools as well as interface with a variety of database management systems. Thisis useful for creating Flask apps that communicate with databases.

15

### CHAPTER – 7 SOFTWARE DESCRIPTION

To Detect the Cancer the needed fields are filled up by the user/Patients where all their lifestyle factors have been given and also possesses User-friendly interface where patients may enter their lifestyle variables. Age, gender,smoking habits, alcohol intake, dietary patterns, family history of cancer, and any other characteristics known to be related with cancer risk should be included in this interface. Validation tests should be implemented to ensure that the submitted data is in the right format and within acceptable limits. Validate areas like age (number), gender (male or female), smoking habits (yes or no), and so forth. If improper data is entered, provide relevant error warnings. Assessment of Lifestyle Factors: Create an algorithm or set of criteria to analyze the influence of each lifestyle factor on cancer risk.

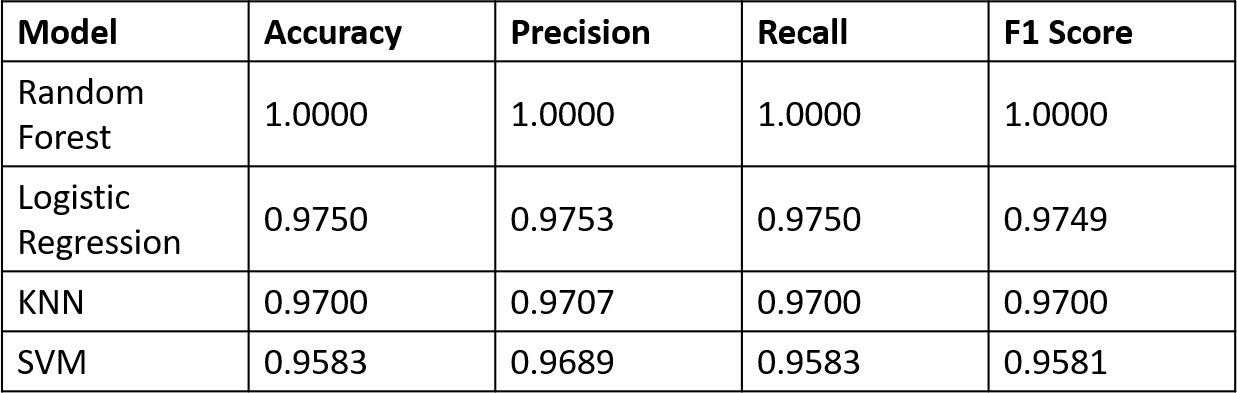
### CHAPTER – 8 RESULTS AND DISCUSSION



##### Fig 8.1 Sample Result

Following Table presents the performance metrics of four machine learning models- Random Forest, Logistic Regression, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM) on the dataset. The performance metrices includes accuracy, precision, recall and F-1scores as they provide a quantitative measure of how well the model is performing well.

### Table: Model Comparison on the dataset



According to the model's accuracy of 1.0 for both the validation and test sets, the random forest model is highlyeffective in predicting cancer based on the provided features. It is clear from the confusion matrices for the validation and test sets that there are no false positives or false negatives in the predictions, demonstrating that the model is capable of correctly predicting the outcomes for all three classes. Each class's accuracy, recall, andf1-score are all 1.0, demonstrating that the model can successfully identify every instance of each class. The successful random forest model implies that the chosen features are incredibly useful in predicting cancer and that the model is able to get the correlations between the features and the direct variable.

Based on these findings, it is possible to construct a more reliable diagnostic tool for cancer detection using the random forest model, which may be useful in detecting cancer using these specific traits. Comparing with the other algorithmsthat are used i.e., Logistic Regression which gives accuracy, precision, recall score of 0.975 and f-1 score of 0.974 which is greater as compared to KNN with 0.97 and SVM with 0.95 scores which gives overall scores lesser than the Random Forest algorithm. The Data in the Dataset is very minimal, but which Random Forest can predict accurately as compared to the other algorithms which concludes that it is more effective and accurateto use it with the dataset and it fits much better than the other algorithms.

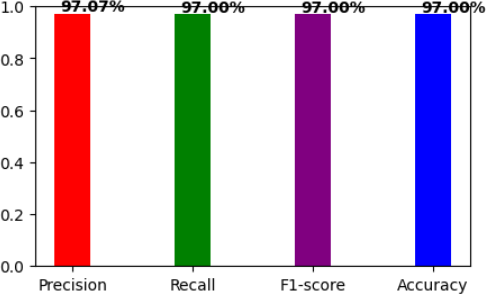
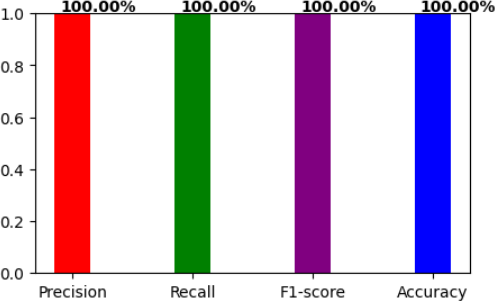
 

Figure 8.2. KNN Results Figure 8.3:CNN Results

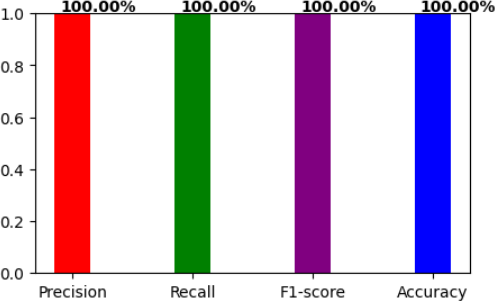
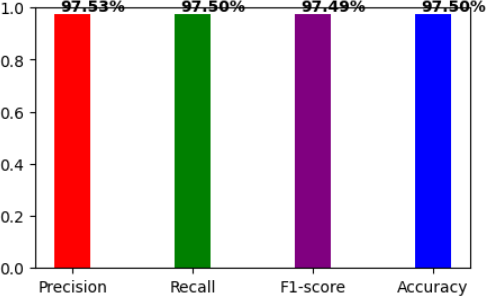


Figure 8.4: Logistic Regression Results Figure 8.5: SVM Results

### CHAPTER – 9 CONCLUSION

In conclusion, the developed machine learning models for predicting cancer risk based on certain features, and found that CNN, Logistic Regression, and RNN performed well among which CNN outperformed all other algorithms used and fits the model best while predicting. This study provides insight into the potential of machine learning models in predicting cancer risk, which can be useful for early detection and treatment. It also handles large data while predicting and thus performed well.

### CHAPTER – 10 FUTURE SCOPE

A Random Forest algorithm was used and adjusted in this work to predict cancer with efficiency and accuracy.However, there are other areas that might be explored and improved in the future. To begin, extending the featureset relevant to cancer can increase the model's prediction skills. Cancer is a complicated illness that is impactedby many factors, and the present model may not have included all essential elements. The model's performance might be enhanced by integrating additional variables such as genetic markers, environmental exposures, lifestyle factors, and biomarkers, resulting in more accurate predictions. Second, the current study made use ofa tiny dataset. A bigger dataset that includes a variety of demographics and cancer types can give more representative and robust training data.

### CHAPTER – 11 REFERENCES

1. Chaurasia V, Pal S, Tiwari B. Prediction of benign and malignant breast cancer using data mining techniques. Journal of Algorithms & Computational Technology. 2018;12(2):119-126. doi:10.1177/1748301818756225
2. IOP Conference Series: Materials Science and Engineering, Volume 1116, International Conference on Futuristic and Sustainable Aspects in Engineering and Technology (FSAET 2020) 18th-19th December 2020, Mathura, India
3. Shafique R, Rustam F, Choi GS, Díez IT, Mahmood A, Lipari V, Velasco CLR, Ashraf I. Breast Cancer Prediction Using Fine Needle Aspiration Features and Upsampling with Supervised Machine Learning. Cancers (Basel). 2023 Jan 22;15(3):681. doi: 10.3390/cancers15030681. PMID: 36765642; PMCID: PMC9913345.
4. Huang T, Le D, Yuan L, Xu S, Peng X. Machine learning for prediction of in-hospital mortality in lung cancer patients admitted to intensive care unit. PLoS One. 2023 Jan 26;18(1):e0280606. doi: 10.1371/journal.pone.0280606. PMID: 36701342; PMCID: PMC9879439.
5. Roberts E, Howell S, Evans DG. Polygenic risk scores and breast cancer risk prediction. Breast. 2023 Feb;67:71-77. doi: 10.1016/j.breast.2023.01.003. Epub 2023 Jan 10. PMID: 36646003; PMCID: PMC9982311.
6. Octaviani, Lidya & Rustam, Zuherman. (2019). Random forest for breast cancer prediction. AIP Conference Proceedings. 2168. 020050. 10.1063/1.5132477.
7. Manas Minnoor, Veeky Baths,Diagnosis of Breast Cancer Using Random Forests, Procedia Computer Science,Volume 218,2023,Pages 429-437,ISSN 1877-

0509,https://doi.org/10.1016/j.procs.2023.01.025.

1. Mohammed Amine Naji, Sanaa El Filali, Kawtar Aarika, EL Habib Benlahmar, Rachida Ait Abdelouhahid, Olivier Debauche,Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis, Procedia Computer Science, Volume 191, 2021, Pages 487-492, ISSN 1877- 0509,https://doi.org/10.1016/j.procs.2021.07.062.([https://www.sciencedirect.com/science/arti](https://www.sciencedirect.com/science/article/pii/S1877050921014629) [cle/pii/S1877050921014629](https://www.sciencedirect.com/science/article/pii/S1877050921014629))
2. Varsha Nemade, Vishal Fegade,Machine Learning Techniques for Breast Cancer Prediction,Procedia Computer Science,Volume 218,2023,Pages 1314-1320, ISSN 1877- 0509,https://doi.org/10.1016/j.procs.2023.01.110.(https://[www.sciencedirect.com/science/arti](http://www.sciencedirect.com/science/arti) cle/pii/S1877050923001102)

22

A.2

1. Random forest for breast cancer prediction Author: Octaviani, T. L.; Rustam, Z. Publication: AIP Conference Proceedings Publisher: AIP Publishing Date: Nov 4, 2019; https://doi.org/10.1063/1.5132477.
2. Mohammed Amine Naji, Sanaa El Filali, Kawtar Aarika, EL Habib Benlahmar, Rachida Ait Abdelouhahid, Olivier Debauche, Machine Learning Algorithms For Breast Cancer Prediction And Diagnosis, Procedia Computer Science, Volume 191, 2021, Pages 487-492, ISSN 1877- 0509,https://doi.org/10.1016/j.procs.2021.07.062.(https://[www.sciencedirect.com/science/arti](http://www.sciencedirect.com/science/arti) cle/pii/S1877050921014629)
3. Kumar, Ajay and Sushil, Rama and Tiwari, Arvind Kumar, Machine Learning Based Approaches for Cancer Prediction: A Survey (March 11, 2019). Proceedings of 2nd International Conference on Advanced Computing and Software Engineering (ICACSE) 2019, Available at SSRN: [https://ssrn.com/abstract=3350294](https://ssrn.com/abstract%3D3350294) or <http://dx.doi.org/10.2139/ssrn.3350294>
4. Predicting metastasis in gastric cancer patients: machine learning-based approaches AtefehTalebi1,2, CarlosA. Celis-Morales2,3, Nasrin Borumandnia 4\*, SomayehAbbasi5,

MohamadAmin Pourhoseingholi6, AbolfazlAkbari7 & JavadYousef

1. Uppara, Rajani & Yadav, Surabhi & Kavitha, Dr.M.. (2023). Voting Classifier on Ensemble Algorithms for Breast Cancer Prediction. 653-660. 10.1109/IDCIoT56793.2023.10053498.
2. Shih, Yang-Hsin & Qin, Gong & Tang, Peifu & Shen, & Liu, Tai-Yi & Gao, Shuai. (2019). Delineation of Urban Growth Boundaries Using a Patch-Based Cellular Automata Model under Multiple Spatial and Socio-Economic Scenarios. Sustainability. 11. 6159. 10.3390/su11216159.
3. <https://dataaspirant.com/how-logistic-regression-model-works/>
4. A Prosthetic Hand Body Area Controller Based on Efficient Pattern Recognition Control Strategies -Scientific Figure on ResearchGate. Available from: https:/[/www.rese](http://www.researchgate.net/figure/SVM-algorithm-)a[rchgate.net/figure/SVM-algorithm-](http://www.researchgate.net/figure/SVM-algorithm-) block-diagram-a-and-memory- allocation-b\_fig3\_316188717

23

A.2

### CHAPTER – 12 APPENDIX

1. **SOURCE CODE:**

import pandas as pd import numpy as np import seaborn as sns import matplotlib.pyplot as plt

# Load the data

data = pd.read\_excel('/content/cancer patient data sets.xlsx')

# Ignore the Patient ID column and extract the features and target variableX = data.iloc[:, 1:-1]

# Compute the correlation matrixcorr\_matrix = X.corr()

# Generate a mask for the upper triangle

mask = np.triu(np.ones\_like(corr\_matrix, dtype=bool))

# Set up the matplotlib figure

fig, ax = plt.subplots(figsize=(10, 10))

# Generate a custom diverging colormap

cmap = sns.diverging\_palette(220, 10, as\_cmap=True)

# Draw the heatmap with the mask and correct aspect ratio sns.heatmap(corr\_matrix, mask=mask, cmap=cmap, vmax=.3, center=0,

square=True, linewidths=.5, cbar\_kws={"shrink": .5}, annot=True, fmt='.2f')

# Rotate the tick labels and set their alignmentplt.xticks(rotation=45, ha="right", fontsize=10)

plt.yticks(fontsize=10)

24

A.2

# Set title and adjust spacing

ax.set\_title("Feature Correlation Matrix", fontsize=16, pad=20) plt.tight\_layout()plt.show()

import pandas as pd import numpy as np import seaborn as sns

# Load the data

data = pd.read\_excel('/content/cancer patient data sets.xlsx')

# Ignore the Patient ID column and extract the features and target variableX = data.iloc[:, 1:-1]

# Compute the correlation matrixcorr\_matrix = X.corr()

# Save the correlation matrix as a CSV filecorr\_matrix.to\_csv('correlation\_matrix.csv')

# Generate a text report with the correlation matrixreport = 'Feature Correlation Matrix:\n\n'

for column in corr\_matrix.columns: corr\_values = corr\_matrix[column].round(2) corr\_values = corr\_values.drop(index=column)

report += f"{column} vs. {corr\_values.idxmax()}: {corr\_values.max()}\n"report += '\n'

# Display the text reportprint(report)

print(data.head())

import pandas as pdimport numpy as np

from sklearn.model\_selection import train\_test\_split from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import confusion\_matrix, classification\_report from sklearn.metrics import accuracy\_score, roc\_auc\_score, roc\_curveimport matplotlib.pyplot as plt

# Load the data

data = pd.read\_excel('/content/cancer patient data sets.xlsx')

# Ignore the Patient ID column and extract the features and target variableX = data.iloc[:, 1:-1]

y = data.iloc[:, -1]

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4, random\_state=0)

# Further split the training data into development and validation sets

X\_dev, X\_val, y\_dev, y\_val = train\_test\_split(X\_train, y\_train, test\_size=0.3, random\_state=0)

#Train and evaluate the Random Forest modelrandom\_forest = RandomForestClassifier() random\_forest.fit(X\_dev, y\_dev)

y\_val\_pred = random\_forest.predict(X\_val) val\_confusion\_matrix = confusion\_matrix(y\_val, y\_val\_pred)

val\_classification\_report = classification\_report(y\_val, y\_val\_pred)y\_test\_pred = random\_forest.predict(X\_test) test\_confusion\_matrix = confusion\_matrix(y\_test, y\_test\_pred)

test\_classification\_report = classification\_report(y\_test, y\_test\_pred)print("Random Forest Results:")

print("Validation Confusion Matrix:") print(val\_confusion\_matrix) print("Validation Classification Report:")print(val\_classification\_report) print("Test Confusion Matrix:") print(test\_confusion\_matrix)

print("Test Classification Report:")print(test\_classification\_report)

models = [random\_forest] model\_names = ['Random Forest']

for model, model\_name in zip(models, model\_names):y\_test\_pred = model.predict(X\_test)

accuracy = accuracy\_score(y\_test, y\_test\_pred)print(f"{model\_name} Accuracy:

{accuracy:.4f}")

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score, precision\_recall\_fscore\_support

import matplotlib.pyplot as pltimport numpy as np

# Define metric names and values

metric\_names = ['Accuracy', 'Precision', 'Recall', 'F1 Score']accuracy = accuracy\_score(y\_test, y\_test\_pred)

precision, recall, f1\_score, \_ = precision\_recall\_fscore\_support(y\_test, y\_test\_pred, average='weighted')

metric\_values = [accuracy, precision, recall, f1\_score]

# Create bar plot

fig, ax = plt.subplots()

x\_pos = np.arange(len(metric\_names))

ax.bar(x\_pos, metric\_values, align='center', alpha=0.5)ax.set\_xticks(x\_pos) ax.set\_xticklabels(metric\_names)ax.set\_ylabel('Value') ax.set\_title('Random Forest Model Metrics')plt.show()

import numpy as np

import matplotlib.pyplot as plt

from sklearn.metrics import confusion\_matrix, classification\_report, accuracy\_score

def colored\_confusion\_matrix(y\_true, y\_pred, labels): cm = confusion\_matrix(y\_true, y\_pred, labels=labels)

cm\_norm = cm.astype('float') / cm.sum(axis=1)[:, np.newaxis] plt.imshow(cm\_norm, interpolation='nearest', cmap=plt.cm.Blues)plt.colorbar()

tick\_marks = np.arange(len(labels)) plt.xticks(tick\_marks, labels, rotation=45) plt.yticks(tick\_marks, labels) plt.xlabel('Predicted Label') plt.ylabel('True Label') plt.title('Confusion Matrix')

plt.tight\_layout()plt.show()

accuracy = accuracy\_score(y\_true, y\_pred)

report = classification\_report(y\_true, y\_pred, labels=labels)return accuracy, report

labels = ['High', 'Medium', 'Low']

accuracy, report = colored\_confusion\_matrix(y\_test, y\_test\_pred, labels) print(f"Accuracy: {accuracy:.4f}")

print(report) import pickle

with open('Cancerdetection.pkl','wb') as f: pickle.dump(model,f)f.close()

with open('Cancerdetection.pkl', 'rb') as file:model = pickle.load(file) y\_train.head(1) answers=[[17,1,3,1,2,5,3,4,2,2,2,2,8,3,3,3,6,2,8,7,5,4,4]]

import pandas as pdimport numpy as np

from sklearn.metrics import confusion\_matrix, classification\_report, precision\_recall\_fscore\_support

from sklearn.model\_selection import train\_test\_split from sklearn.ensemble import RandomForestClassifierfrom sklearn.linear\_model import LogisticRegression from sklearn.neighbors import KNeighborsClassifier from sklearn.svm import SVC

from sklearn.metrics import confusion\_matrix, classification\_report from sklearn.metrics import accuracy\_score, roc\_auc\_score, roc\_curveimport matplotlib.pyplot as plt

# Load the data

data = pd.read\_excel('/content/cancer patient data sets.xlsx')

# Ignore the Patient ID column and extract the features and target variableX = data.iloc[:, 1:-1]

y = data.iloc[:, -1]

# Split the data into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.4, random\_state=0)

# Further split the training data into development and validation sets

X\_dev, X\_val, y\_dev, y\_val = train\_test\_split(X\_train, y\_train, test\_size=0.3, random\_state=0)

# Train and evaluate the Random Forest model

random\_forest = RandomForestClassifier(n\_estimators=100, max\_depth=5, random\_state=0)random\_forest.fit(X\_dev, y\_dev)

y\_val\_pred = random\_forest.predict(X\_val) val\_confusion\_matrix = confusion\_matrix(y\_val, y\_val\_pred)

val\_classification\_report = classification\_report(y\_val, y\_val\_pred)y\_test\_pred = random\_forest.predict(X\_test) test\_confusion\_matrix = confusion\_matrix(y\_test, y\_test\_pred)

test\_classification\_report = classification\_report(y\_test, y\_test\_pred)

print("Random Forest Results:") print("Validation Confusion Matrix:") sns.heatmap(val\_confusion\_matrix, annot=True, cmap="Blues", fmt="d")plt.show() print("Validation Classification Report:")print(val\_classification\_report) print("Test Confusion Matrix:")

sns.heatmap(test\_confusion\_matrix, annot=True, cmap="Blues", fmt="d")plt.show() print("Test Classification Report:")print(test\_classification\_report)

# Plot the colored confusion matrix

cm = confusion\_matrix(y\_test, y\_test\_pred)plt.figure(figsize=(6, 6)) sns.heatmap(cm, annot=True, cmap="YlGnBu", fmt="d", xticklabels=['Benign', 'Malignant'],yticklabels=['Benign', 'Malignant'])

plt.title('Random Forest - Confusion Matrix')plt.ylabel('Actual') plt.xlabel('Predicted')plt.show()

# Train and evaluate the Logistic Regression modellogreg = LogisticRegression() logreg.fit(X\_dev, y\_dev) y\_val\_pred = logreg.predict(X\_val)

val\_confusion\_matrix = confusion\_matrix(y\_val, y\_val\_pred) val\_classification\_report = classification\_report(y\_val, y\_val\_pred)y\_test\_pred = logreg.predict(X\_test) test\_confusion\_matrix = confusion\_matrix(y\_test, y\_test\_pred) test\_classification\_report = classification\_report(y\_test, y\_test\_pred)print("Logistic Regression Results:")

print("Validation Confusion Matrix:")

sns.heatmap(val\_confusion\_matrix, annot=True, cmap="Blues", fmt="d")plt.show() print("Validation Classification Report:")print(val\_classification\_report) print("Test Confusion Matrix:")

sns.heatmap(test\_confusion\_matrix, annot=True, cmap="Blues", fmt="d")plt.show() print("Test Classification Report:")print(test\_classification\_report)

# Train and evaluate the KNN model

knn = KNeighborsClassifier(n\_neighbors=5)knn.fit(X\_dev, y\_dev) y\_val\_pred = knn.predict(X\_val)

val\_confusion\_matrix = confusion\_matrix(y\_val, y\_val\_pred) val\_classification\_report = classification\_report(y\_val, y\_val\_pred)y\_test\_pred = knn.predict(X\_test) test\_confusion\_matrix = confusion\_matrix(y\_test, y\_test\_pred) test\_classification\_report = classification\_report(y\_test, y\_test\_pred)print("KNN Results:")

print("Validation Confusion Matrix:")

sns.heatmap(val\_confusion\_matrix, annot=True, cmap="Blues", fmt="d")plt.show() print("Validation Classification Report:")print(val\_classification\_report) print("Test Confusion Matrix:")

sns.heatmap(test\_confusion\_matrix, annot=True, cmap="Blues", fmt="d")plt.show() print("Test Classification Report:")print(test\_classification\_report)

models = [random\_forest, logreg, knn]

model\_names = ['Random Forest', 'Logistic Regression', 'KNN']

for model, model\_name in zip(models, model\_names):y\_test\_pred = model.predict(X\_test)

precision, recall, fscore, support = precision\_recall\_fscore\_support(y\_test, y\_test\_pred,average='weighted')

accuracy = accuracy\_score(y\_test, y\_test\_pred)print(f"{model\_name} Results:") print(f"Accuracy: {accuracy:.4f}") print(f"Precision: {precision:.4f}") print(f"Recall: {recall:.4f}")

print(f"F1-score: {fscore:.4f}")print("")

# Plot the precision, recall, accuracy, and F1-scoreplt.figure(figsize=(5,3)) plt.bar(['Precision', 'Recall', 'F1-score', 'Accuracy'], [precision, recall, fscore,

accuracy],color=['red', 'green', 'purple', 'blue'],width=0.3) plt.ylim([0, 1])

plt.title(f"{model\_name} Performance Metrics\n\n")

for i, v in enumerate([precision, recall, fscore, accuracy]): plt.text(i-0.1, v+0.01, f"{v:.2%}",color='black',fontweight='bold')

plt.show()

# Initialize the SVM model with regularization

svm\_model = SVC(C=0.36, kernel='rbf', gamma='scale', class\_weight='balanced')

# Fit the model on the development datasvm\_model.fit(X\_dev, y\_dev)

# Evaluate the model on the validation sety\_val\_pred = svm\_model.predict(X\_val) val\_confusion\_matrix = confusion\_matrix(y\_val, y\_val\_pred) val\_classification\_report = classification\_report(y\_val, y\_val\_pred)print("SVM Results:")

print("Validation Confusion Matrix:") print(val\_confusion\_matrix) print("Validation Classification Report:")print(val\_classification\_report)

# Tune hyperparameters on the validation setc\_values = [0.1, 1, 10]

kernel\_values = ['linear', 'rbf', 'poly']gamma\_values = ['scale', 'auto'] best\_accuracy = 0 best\_c = 0 best\_kernel = '' best\_gamma = ''for c in c\_values:

for kernel in kernel\_values:

for gamma in gamma\_values:

svm\_model = SVC(C=c, kernel=kernel, gamma=gamma, class\_weight='balanced') svm\_model.fit(X\_dev, y\_dev)

y\_val\_pred = svm\_model.predict(X\_val) accuracy = accuracy\_score(y\_val, y\_val\_pred) if accuracy > best\_accuracy:

best\_accuracy = accuracybest\_c = c best\_kernel = kernel best\_gamma = gamma

# Train the SVM model on the entire training set with the best hyperparameters svm\_model = SVC(C=best\_c, kernel=best\_kernel, gamma=best\_gamma,

class\_weight='balanced')svm\_model.fit(X\_train, y\_train)

# Evaluate the model on the test set y\_test\_pred = svm\_model.predict(X\_test) test\_confusion\_matrix = confusion\_matrix(y\_test, y\_test\_pred) test\_classification\_report = classification\_report(y\_test, y\_test\_pred)print("Test Confusion Matrix:")

print(test\_confusion\_matrix) print("Test Classification Report:") print(test\_classification\_report)

svm\_model = SVC(C=best\_c, kernel=best\_kernel, gamma=best\_gamma, class\_weight='balanced')svm\_model.fit(X\_train, y\_train)

# Evaluate the model on the test set y\_test\_pred = svm\_model.predict(X\_test) precision, recall, fscore, support = precision\_recall\_fscore\_support(y\_test, y\_test\_pred, average='weighted')

accuracy = accuracy\_score(y\_test, y\_test\_pred)

print("SVM Results:") print(f"Accuracy: {accuracy:.4f}")print(f"Precision: {precision:.4f}") print(f"Recall: {recall:.4f}")

print(f"F1-score: {fscore:.4f}")print("")

# Plot the precision, recall, accuracy, and F1-scoreplt.figure(figsize=(5,3))

plt.bar(['Precision', 'Recall', 'F1-score', 'Accuracy'], [precision, recall, fscore, accuracy], color=['red','green', 'purple', 'blue'],width=0.3)

plt.ylim([0, 1])

plt.title("SVM Performance Metrics\n\n")

for i, v in enumerate([precision, recall, fscore, accuracy]): plt.text(i-0.1, v+0.01, f"{v:.2%}",color='black',fontweight='bold')

plt.show()

import matplotlib.pyplot as pltimport seaborn as sns

# Plot the confusion matrix for the validation setsns.set(font\_scale=1.4) sns.heatmap(val\_confusion\_matrix, annot=True, annot\_kws={"size": 16}, cmap="Blues", fmt='g')plt.title("Validation Confusion Matrix") plt.xlabel("Predicted Class")plt.ylabel("True Class") plt.show()

# Plot the confusion matrix for the test setsns.set(font\_scale=1.4) sns.heatmap(test\_confusion\_matrix, annot=True, annot\_kws={"size": 16}, cmap="Blues", fmt='g')plt.title("Test Confusion Matrix") plt.xlabel("Predicted Class")plt.ylabel("True Class") plt.show()

from sklearn.ensemble import RandomForestClassifierfrom sklearn.linear\_model import LogisticRegression from sklearn.neighbors import KNeighborsClassifier

from sklearn.svm import SVC

from sklearn.metrics import accuracy\_scoreimport matplotlib.pyplot as plt

random\_forest = RandomForestClassifier(n\_estimators=100, max\_depth=5, random\_state=0)random\_forest.fit(X\_dev, y\_dev)

y\_val\_pred\_rf = random\_forest.predict(X\_val) val\_accuracy\_rf = accuracy\_score(y\_val, y\_val\_pred\_rf) y\_test\_pred\_rf = random\_forest.predict(X\_test) test\_accuracy\_rf = accuracy\_score(y\_test, y\_test\_pred\_rf)

logreg = LogisticRegression() logreg.fit(X\_dev, y\_dev) y\_val\_pred\_lr = logreg.predict(X\_val)

val\_accuracy\_lr = accuracy\_score(y\_val, y\_val\_pred\_lr)y\_test\_pred\_lr = logreg.predict(X\_test)

test\_accuracy\_lr = accuracy\_score(y\_test, y\_test\_pred\_lr)

knn = KNeighborsClassifier(n\_neighbors=5)knn.fit(X\_dev, y\_dev) y\_val\_pred\_knn = knn.predict(X\_val)

val\_accuracy\_knn = accuracy\_score(y\_val, y\_val\_pred\_knn)y\_test\_pred\_knn = knn.predict(X\_test)

test\_accuracy\_knn = accuracy\_score(y\_test, y\_test\_pred\_knn)

svm\_model = SVC(C=best\_c, kernel=best\_kernel, gamma=best\_gamma, class\_weight='balanced')svm\_model.fit(X\_dev, y\_dev)

y\_val\_pred\_svm = svm\_model.predict(X\_val) val\_accuracy\_svm = accuracy\_score(y\_val, y\_val\_pred\_svm) y\_test\_pred\_svm = svm\_model.predict(X\_test) test\_accuracy\_svm = accuracy\_score(y\_test, y\_test\_pred\_svm)

models = ['Random Forest', 'Logistic Regression', 'KNN', 'SVM'] val\_accuracies = [val\_accuracy\_rf, val\_accuracy\_lr, val\_accuracy\_knn, val\_accuracy\_svm] test\_accuracies = [test\_accuracy\_rf, test\_accuracy\_lr, test\_accuracy\_knn, test\_accuracy\_svm]

fig, axs = plt.subplots(2, figsize=(10, 10))

axs[0].bar(models, val\_accuracies, color=['orange', 'green', 'blue', 'red'], width=0.3) axs[0].set\_title('Validation Accuracy of Different Models', pad=20) axs[0].set\_xlabel('Model')

axs[0].set\_ylabel('Accuracy')axs[0].set\_ylim([0, 1])

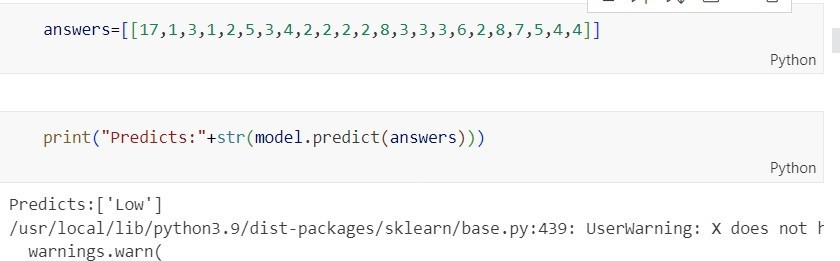
axs[1].bar(models, test\_accuracies, color=['orange', 'green', 'blue', 'red'], width=0.3) axs[1].set\_title('Test Accuracy of Different Models', pad=20) axs[1].set\_xlabel('Model') axs[1].set\_ylabel('Accuracy')axs[1].set\_ylim([0, 1])

for ax in axs:

for i, v in enumerate(val\_accuracies):

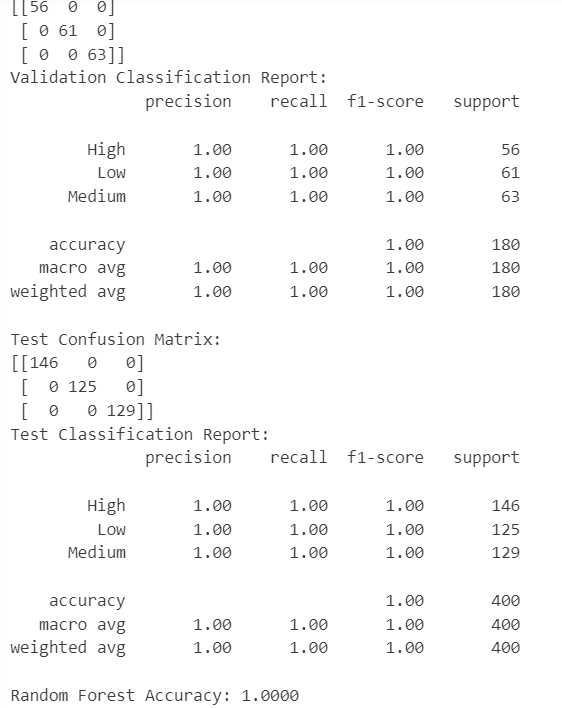
ax.text(i-0.15, v+0.02, str(round(v\*100, 2))+'%', color='black', fontweight='bold') plt.tight\_layout()plt.show()

### SCREENSHOTS Testing Pickle Model:



**Fig B.1.1 Sample test for pickle model**

### Model Trained with accuracy:

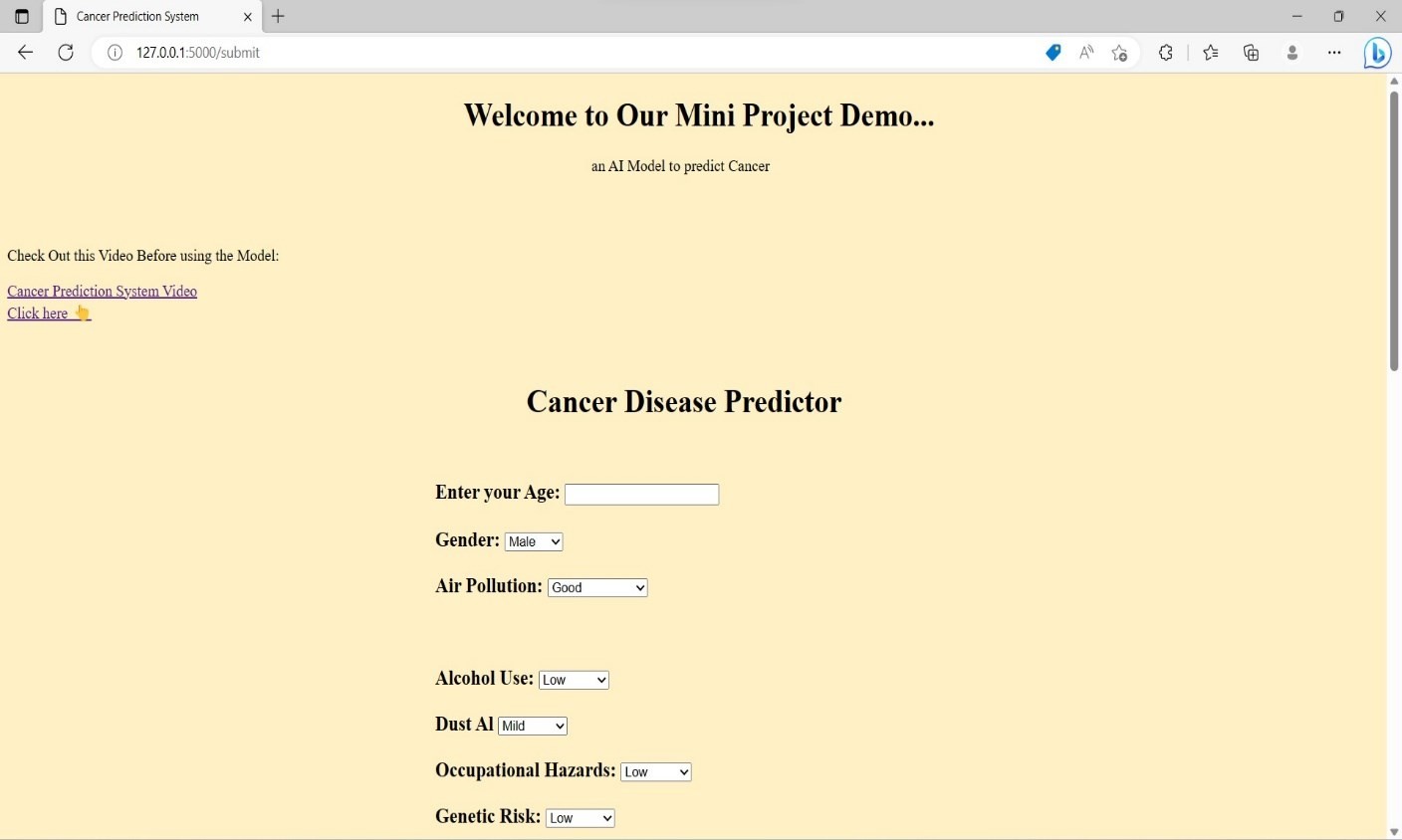


##### Fig B.1.2 Trained Data and its accuracies with metrices

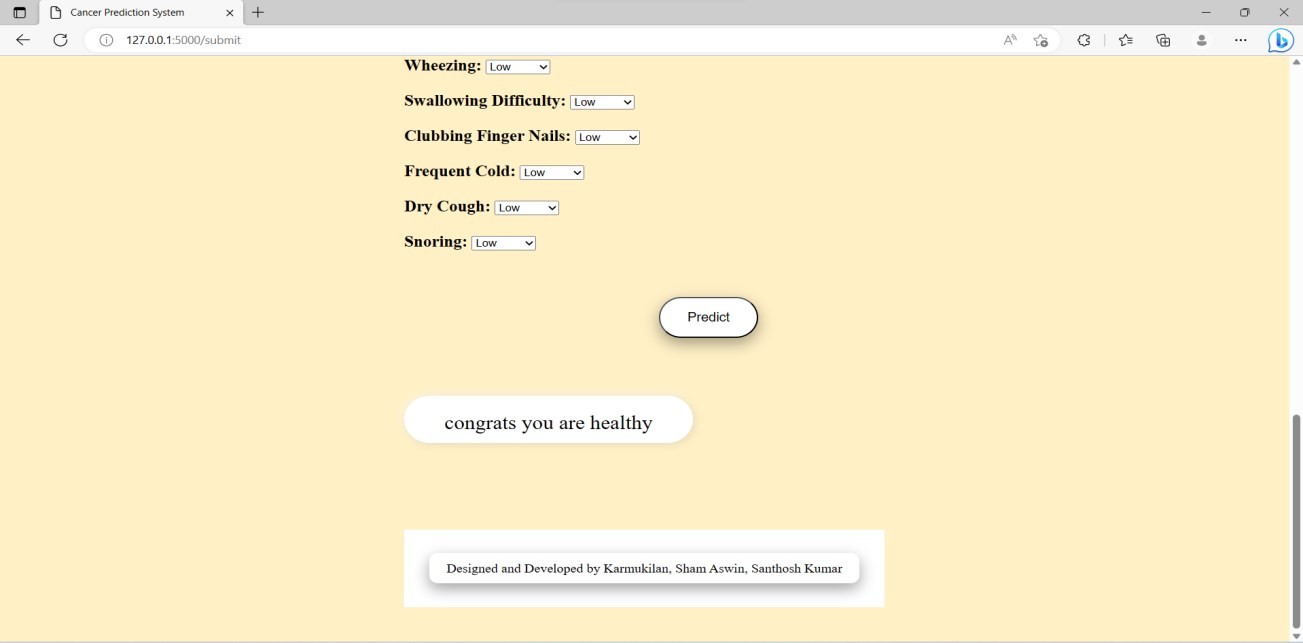
37

B.1

### Webpage



##### Fig B.1.3 Developed web page



**Fig B.1.4 Output**

B.2