BLOOD CANCER DETECTION USING MACHINE LEARNING

Final report of the minor project is being submitted for partial fulfillment of the requirements for the degree of

Bachelor in Computer Application **(***BCA)*

from Maulana Abul Kalam Azad University of Technology, West Bengal

Student of BCA– fifth Semester

Academic Session: 2022 - 2023

by

PRERNA KUMARI SHAW
Registration No: 213771001210014 OF 2021-22
University Roll No: 37701221005
&

TUNIR DUTTA
Registration No: 213771001210009 OF 2021-22
University Roll No: 37701221013

Student of BCA – Fifth Semester

Academic Session: 2023 - 2024

Under the supervision of

Mr. Falguni Adhikary

at

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Canal South Road, Beleghata, Kolkata, 700015

Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal

(Erstwhile WBUT)

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Canal South Road, Beliaghata, Kolkata, 700015

Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal

(Erstwhile WBUT)

FORWARD

The report of the minor project titled "***Blood Cancer Detection Using Machine Learning"*** submitted by Prerna Kumari Shaw & Tunir Dutta, of BCA 5thSemester, Academic session – 2023 – 2024, have been prepared under my supervision for the partial fulfillment of the requirements for B.SC in Computer Science degree in Maulana Abul Kalam Azad University of Technology, W.B. (erstwhile, WBUT).

The report is hereby forwarded.

Date: *23rd November,2023*

Falguni Adhikary
Assistant Professor
Department of CS, Non AICTE
RCCIIT, Kolkata (Supervisor)

ACKNOWLEDGEMENT

I express my sincere gratitude to my supervisor Mr. Falguni Adhikary for extending his tremendous support and guiding me to take up this problem as my minor project report. I have been immensely benefited by his continuous follow up, rigorous monitoring and technical support.

I am also indebted to Dr. Anirban Mukherjee, Principal (*In-Charge)* RCCIIT and Dr. Arindam Mandal (HOD, Dept. of CS) for guiding me at my initial stage of project and arranging necessary infrastructural support from the Department of CS (Non AICTE).

It is needless to mention yet I want to express my sincere gratitude to my other faculty members of the Department and our project coordinator, Ms. Sudipta Dey for helping me all along the tenure of my study.

I am thankful to Mr. Swarup Kumar Paul & all other respected faculty members from my department for their inspiring words and support all along the study tenure.

I must thank my classmates for their continuous support and all of them were very kind and helpful, hence it is difficult to pick any particular name for special acknowledgement.

Date: 23rd November, 2023

_____

Prerna Kumari Shaw

(Reg no: **213771001210014,**

Roll no: 37701221005)

Tunir Dutta

(Reg no: **213771001210009,**

Roll no: 37701221013)

RCC INSTITUTE OF INFORMATION TECHNOLOGY

Canal South Road, Beliaghata, Kolkata, 700015

Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal

(Erstwhile WBUT)

Accredited by AICTE, New Delhi, India

CERTIFICATE of ACCEPTANCE

*The Minor Project titled ""**Blood Cancer Detection Using Machine Learning"*** submitted by Prerna Kumari Shaw & Tunir Dutta*, of BCA 5thSemester (Academic session: 2023 – 2024) is hereby recommended to be accepted for the partial fulfillment of the requirements for BCA degree from Maulana Abul Kalam Azad University of Technology, West Bengal (formerly known as WBUT)*

Name of the Examiner	***Signature with Date***
1. ……	……..….
2. ……	……..
3. ……	……
4. ……..	……

CONTENTS

1. INTRODUCTION

Blood cancer, also known as hematologic cancer, it is a broad category of highly differentiated cancers that impact the bone marrow, which produces blood cells, as well as the blood itself. Different blood cell types, including white blood cells, red blood cells, and platelets, are the source of these tumors. Among the most prevalent types of blood cancers are lymphoma and myeloma. DNA alterations within blood cells are the root cause of blood cancer.

Our proposed model represents a significant advancement in the field of healthcare and cancer detection. At its core, this model is designed to predict the presence of blood cancer by assessing a comprehensive set of parameters related to normal blood cells, primarily focusing on white blood cells, red blood cells, and platelets. These parameters are fundamental in understanding the overall health and composition of an individual's blood.

We intend to achieve this by leveraging machine learning algorithms, specifically Random Forest (RF), k-Nearest Neighbors (KNN), and Support Vector Machines (SVM). These algorithms will be employed to analyze comprehensive data pertaining to various parameters of normal blood cells, including white blood cells, red blood cells, and platelets. By harnessing the power of data-driven analysis, we aim to identify potential cases of blood cancer at an earlier, more treatable stage.

1.1 APPLICATION

In the application phase of our project aimed at improving early detection of blood cancer using predictive modeling, we embark on practical implementations that directly impact healthcare delivery and patient outcomes:
Our validated predictive model is seamlessly integrated into existing healthcare systems or developed as a standalone application/tool. This assists clinicians by providing them with an efficient means to assess the likelihood of blood cancer based on patient-specific data.

Collaborating closely with healthcare institutions, we spearhead early screening programs that leverage our predictive model. These programs specifically target high-risk populations or individuals exhibiting suspicious symptoms, aiming to detect potential cases of blood cancer at the earliest possible stage.

Employing our predictive model, we stratify patients according to their risk of developing blood cancer. This personalized approach facilitates tailored monitoring and allows for proactive intervention strategies, optimizing patient management.

Innovation extends to telemedicine platforms where our predictive model is integrated. By facilitating remote blood cancer risk assessment and monitoring, particularly for underserved or remote populations, we strive to broaden access to early detection methods.

1.2 MOTIVATION

The motivation behind this project is deeply rooted in the pressing need for advancements in blood cancer diagnostics, which has served as the driving force propelling our research and development efforts. This initiative recognizes the gravity of the situation concerning blood cancer and its impact on public health, offering a comprehensive response to a significant global health challenge. By focusing on improving early detection techniques, we aim to make a substantial and positive contribution to the well-being of individuals affected by blood cancer worldwide.

Blood cancer is a formidable adversary that affects a substantial fraction of the global population. The sheer prevalence of blood cancer underscores the urgency of developing more effective diagnostic methods. The potential for enhancing early detection capabilities is not only a matter of improving individual patient outcomes but also holds the promise of substantially improving public health. By identifying cases of blood cancer at earlier, more manageable stages, we have the opportunity to mitigate the suffering and loss associated with this disease on a large scale.

Our project's focus on machine learning provides a unique learning opportunity for individuals and institutions involved. It encourages the exploration of novel techniques and methodologies, fostering a culture of innovation and adaptability within the healthcare and data science communities. This interdisciplinary approach underscores the importance of collaboration between data scientists, medical professionals, and researchers. By working together, we can

harness the full potential of both data science and medical expertise to address complex medical challenges.

In summary, this project is a complex response to a real global health concern rather than just an academic or technological exercise. It integrates the need to treat blood cancer as soon as possible, the strength of data-driven solutions, the potential for interdisciplinary cooperation, and the moral obligation to enhance patient care. With this program, we hope to significantly improve the lives of people impacted by blood cancer and foster a more compassionate and healthy global community.

1.3 PROBLEM ANALYSIS

In this project, the primary objective is to develop a machine learning model that predicts whether a patient is positive or negative for blood cancer based on eight parameters: patient IDs, body temperature, weight records, White Blood Cell (WBC) Counts, cancer cell Counts, cell sizes, and size of lumps.

The initial phase involves acquiring and understanding the dataset. Gathering comprehensive datasets containing these parameters is crucial. Understanding the data sources, formats, and potential biases or missing values within the dataset is essential for subsequent analysis.

The next step encompasses exploratory data analysis (EDA). This involves delving into the dataset to comprehend the distributions, relationships, and characteristics of each parameter. Visualization techniques like scatter plots, histograms, or correlation matrices can reveal potential patterns or associations between parameters.

Following EDA, the problem statement needs refinement. Clarifying whether the task is a binary classification problem (positive/negative for blood cancer) or a probability estimation for blood cancer risk assists in determining the appropriate modeling approach.

Feature engineering and selection come into play to assess the relevance of each parameter in predicting blood cancer. Understanding feature importance and correlations with the target variable aids in determining which parameters significantly contribute to the predictive power of the model

Data preprocessing steps, such as handling missing values, outliers, and normalization of features, ensure that the data is prepared for modeling. Addressing these issues enhances the model's accuracy and reliability.

Moving to the core of the project, appropriate machine learning algorithms (e.g., Random Forest, Logistic Regression, Support Vector Machines) are selected based on the nature of the problem (classification) and dataset characteristics. These algorithms are trained using the prepared dataset, with a portion reserved for validation or testing purposes.

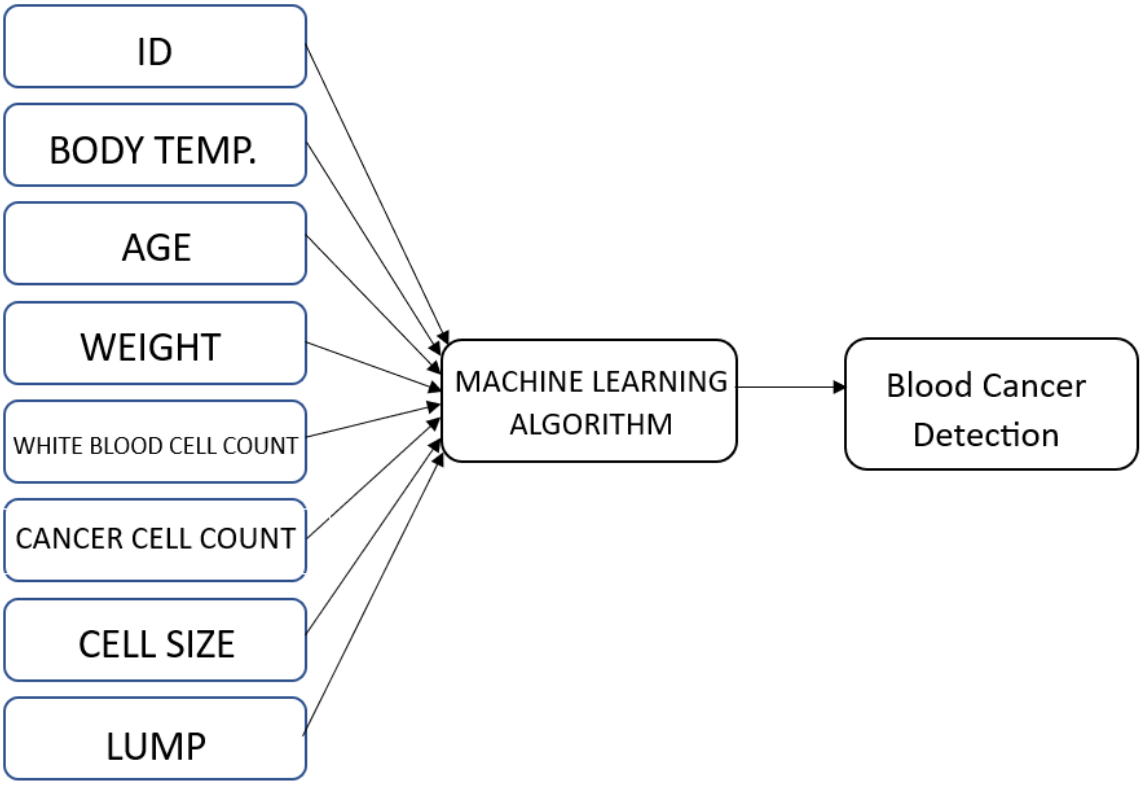

Fig- 1.1: Problem analysis

RF, SVM, KNN is introduced to analyze from our dataset for training & testing and to predict the blood cancer.

1.4 WORK FLOW

Step 1: Firstly we collect the input data considering the parameters such as patient IDs, body temperature, weight records, White Blood Cell (WBC) Counts, cancer cell Counts, cell sizes, and size of lumps.

Step 2: Then datasets are preprocessed on collected data. The main goal of this step is to whether the patient is positive or negative of Blood Cancer.

Step 3: Important features are to be extracted in these steps.

Step 4: Preliminary data is then analyzed using different data analysis methods. Appropriate algorithms should be used to provide crop recommendations.

Step 5: Finally, it is presented as an output i.e. the patient is positive in Blood Cancer or not.

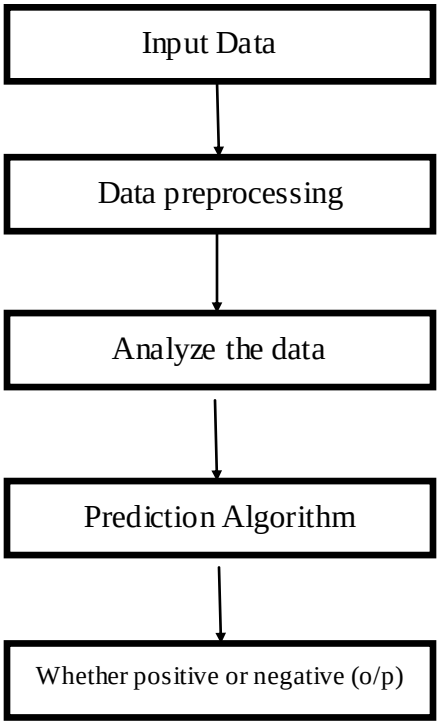

Fig- 1.2: Work flow diagram

2. LITERATURE STUDY

Saranyan, et al. in their 2021 paper, presented that Machine learning methods are currently being used in medical fields for identification and classification of blood cancer disease in early stages . Leukemia is a blood cancer caused by abnormal White Blood Cell (WBC) growth. Historically, diagnosing diseases and Counting blood cells required a lot of manual labor and instruments like the hemocytometer, which produced labor-intensive and inaccurate findings.

The authors suggested a computer-based alternative that makes use of medical image processing methods. This technique, which accurately Counts Red Blood Cells (RBCs), White Blood Cells (WBCs), and Platelets by examining microscopic images, makes it possible to identify suspected leukemia cases in addition to cell Counts. This development provides a more effective and precise approach to disease detection and cell Counting in compared to other manual methods.

- Puneet , Anamika Chauhan. “ Supervised detection of Lung cancer using Machine learning techniques based on routine blood indices” In 2020 IEEE International Conference for Innovation in Technology (INOCON)Bengaluru, India. Nov 6-8, 2020.
- R. Duggal, A. Gupta, and R. Gupta, "Segmentation of overlapping/touching white blood cell nuclei using artificial neural networks" CME Series on Hemato-Oncopathology. All India Institute of Medical Sciences (AIIMS), New Delhi, India, July 2016.
- S. Kumar, S. Mishra, P. Asthana," Automated Detection of AcuteLeukemia Using K-mean Clustering Algorithm". InAdvances inComputer and Computational Sciences. 2018, pp. 655-670
- Tomasz Markiewicz, Stanislaw Osowski, Bonenza Marianska, and Leszek Moszczynski. Automatic recognition of the blood cells of myelogenous leukemia using svm. In Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., volume 4, pages 2496-2501.IEEE, 2005

3. WORKING ALGORITHMS

In the project focused on improving early detection of blood cancer, three primary machine learning algorithms—Support Vector Machines (SVM), Random Forest, and k-Nearest Neighbors (KNN)—play pivotal roles in the predictive modeling phase.

Random Forest (RF):

Random Forest is a learning technique that operates by constructing multiple decision trees during the training phase and merging their predictions to obtain more accurate and robust results. Each decision tree in the Random Forest is built using a random subset of features and training data. In the blood cancer detection project, Random Forest is employed to classify patients into positive or negative blood cancer categories based on their parameters. The nature of Random Forest helps mitigate overfitting and enhances predictive accuracy by aggregating predictions from multiple trees.

Support Vector Machines (SVM):

SVM is a powerful supervised learning algorithm used for both classification and regression tasks. In the context of blood cancer detection, SVM is utilized as a binary classifier to predict whether a patient is positive or negative for blood cancer based on the provided parameters. SVM works by finding an optimal hyperplane that best separates the data points into different classes, maximizing the margin between the classes while minimizing classification errors. The algorithm can handle complex datasets by using kernel functions to transform the input data into higher-dimensional spaces, making it effective in handling non-linear relationships between features.

K-Nearest Neighbors (KNN):

The k-Nearest Neighbors (KNN) algorithm is a machine learning method used for classification tasks, such as predicting blood cancer status based on patient parameters. In the context of this project, KNN operates by comparing new data points with existing labeled data to make predictions

3.1 RANDOM FOREST

The classification, regression and other tasks can also do by RF. It creates different decision trees. That is divided into classification & regression phases. The algorithm creates multiple decision trees by randomly selecting subsets of the data and features to use at each node of the tree. The main idea of this algorithm is to prepare several decision trees, collect their predictions and predict the final result by using multivariate statistics in the case of classification or averaging in the case of regression.

The final prediction is made by aggregating the predictions of all the trees. Then estimate the performance of the trained model on the testing set. Integrate the trained RF model with our recommendation model. The accuracy depends on used tree. Lastly takes parameters as input from diagnostic center and apply the algorithms based on trained model. Here we take the inputs of eight parameter from our dataset & detected the Blood Cancer.

As RF is a powerful ML algorithm. However, the new data points tend to overlap with training data set, leading to incorrect predictions. In summary, fully developed decision trees often provide low-bias and high-variance models where it needed. RF transforms a low-bias, high-variance model into a low-bias low-variance model by training multiple decision trees simultaneously. Each decision tree in the RF takes a subset of the training data set and predicts the result accordingly.

The RF then collects those results and performs various operations to produce the final prediction.

RF is a powerful algorithm that can handle a large number of different ideas as we have 500 in our database. It is very effective because it relies on a set of decision trees where each tree is constructed from a random subset of the input properties. It is necessary to agree that the patient is positive or negative.

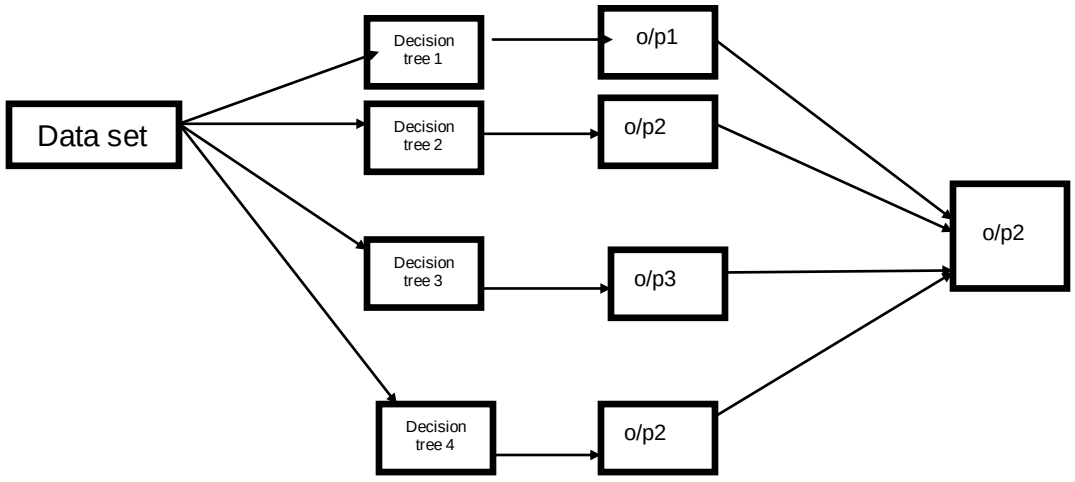

Fig- 3.1: Working of a random-forest algorithm

3.2 SUPPORT VECTOR MACHINE

SVMs are often represented in space as a training data process segmented into groups according to an incomprehensible length. In SVM technique each value represents a coordinate system of values. Classification is done by determining which hyper plane best separates the two classes. SVM generates large lines or planes in high or infinite space for other tasks such as classification, regression, or anomaly detection. SVM allows data to be separated with maximum width. Support vector is the data point that is farthest from the decision surface (or hyper plane). They directly affect the decisions.

Distance of nearest data point from two sets is called the margin. The goal is largest distance selection between the hyper plane and any point in training set, giving it the best chance of classifying the new data. SVM is used as recommendation system. There you have a very high level introduction to SVM.

The SVM is useful in high-speed environments. The SVM works well when number of dimensions is greater than number of instances. The SVM is memory efficient as the vector classifier works by putting data points, above and below the classifying hyper plane. Because support vector classifiers work by placing points above and below the distribution hyper plane, the distribution has no definition.

SVM is used as

- Data Collection**:**
  Patient data such as body temperature, weight records, White Blood Cell (WBC) Counts, cancer cell Counts, cell sizes, and size of lumps are collected.
- Data processing**:** It involves correcting errors, remove duplicates, filling in missing values.
- Feature extraction: In this step features are extracted.
- Model training: Train the model on extracted objects using training and validation methods.
- Model Evaluation: Evaluate performance of the SVM model in a separate test using metrics.
- Model Optimization**:** Optimize SVM model by choosing different kernels such as linear, polynomial or radial basis functions.

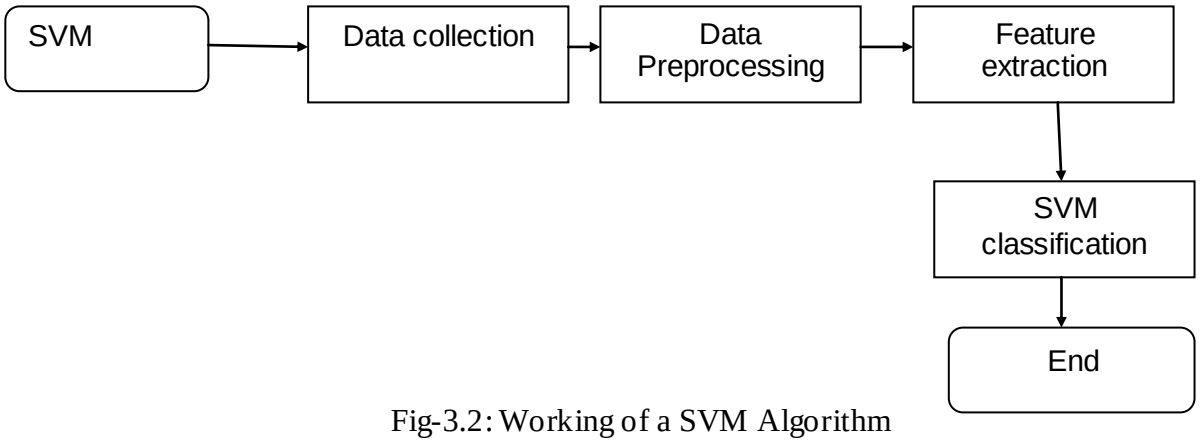

Fig-3.2: Working of a SVM Algorithm

3.3 KNN ALGORITHM

The k-Nearest Neighbors (KNN) algorithm is a machine learning method used for classification tasks, such as predicting blood cancer status based on patient parameters. In the context of this project, KNN operates by comparing new data points with existing labeled data to make predictions.

During the training phase, KNN stores the dataset comprising patient parameters and their corresponding blood cancer status labels. When a new patient's data is provided for prediction, KNN identifies the "k" nearest neighbors in the training dataset, considering similarity in parameter values.

The predictive mechanism of KNN relies on majority voting among the nearest neighbors: the class label most prevalent among the "k" neighbors becomes the predicted class for the new patient. For instance, if the majority of the nearest neighbors have a positive blood cancer status, KNN predicts a positive status for the new patient.

In the specific code context, the dataset is split into training and testing sets. Features are standardized to ensure consistent scaling using StandardScaler. The KNN classifier is then created and trained on the standardized training data. Subsequently, predictions are made on the standardized test data using this classifier.

Evaluating the KNN model involves comparing the predicted blood cancer status with the actual status from the test dataset. The accuracy of these predictions is calculated using a metric such as accuracy_score, which quantifies the correctness of the predicted outcomes.
It's essential to note that the selection of the hyperparameter "k" significantly influences the model's performance. The choice of an optimal "k" value ensures the model does not overfit or underfit the data. Additionally, KNN's computational complexity can increase with larger datasets due to its need to compare new data points with all existing data points during prediction.

Despite its simplicity, KNN can be a powerful tool for classification tasks, particularly in scenarios where the decision boundary is not well defined or in handling non-linear data. In the context of blood cancer detection, KNN offers an intuitive approach to predict blood cancer status based on patient parameters, contributing to early detection efforts in healthcare.

4. PROBLEM DESIGN AND IMPLEMENTATION

Collecting the information about the patient who show the symptoms of blood cancer or who has higher risk of contracting the disease. This information may be gathered from surveys, hospitals, diagnostic center and other sources. The data must first be processed to ensure it is clean and ready.
Next step is to choose an appropriate ML algorithm for prediction. This may depend on the characteristics of the data. Our models use crop recommendation algorithms like KNN, SVM, and RF algorithms.

The next step is to train the model on preprocessed data. Once the model is trained, it needs to be evaluated to ensure the accuracy & reliability. This includes testing the model on test data and comparing its predictions with training data.
After the model is evaluated and determined to be correct and reliable in real situation. Now the doctors can have access to their patients details and make beforehand treatment.

4.1 DATASET AND ENVIRONMENT SELECTION

Our dataset has more than 500 data with eight parameters which are used in the dataset.

The parameters used here which is to be predicted as object data type. Datasets represents different types of symptoms found in the patients in India.

#	Column	Non-Null Count	Dtype
0	ID	499 non-null	int64
1	Body temp ()	499 non-null	float64
2	Age	499 non-null	int64
3	Weight(Kg)	499 non-null	float64
4	WBC Count (per mm3)	499 non-null	int64
5	Cancer Cell Count	499 non-null	int64
6	Cell size (Micrometer)	499 non-null	float64
7	Lump (Millimeter)	499 non-null	float64
8	Result	499 non-null	object

Fig-4.1: Parameters of Dataset.

Here patient ID , age, WBC Count, cancer cell Count are int type & remaining 4 parameters are float data type. In our dataset we are considering different parameters of the patients and predicting that the patient is positive or negative of blood cancer or not.

Data size: Dataset size is also very important. Bigger agricultural dataset can improve the performance of algorithms by providing more information

- Data quality: Data should be good with less error or error free. Data must be cleaned and preprocessed before use to remove duplicate data, incorrect data.

The data are collected from our work team members, survey data, and website and kaggle. This dataset contains yield for major symptoms of suspected patients across all India. The details of parameter of our dataset as follows

Sl No.	Feature	Description
1	ID	Patient's ID
2	Body temp	Body temperature of patient
3	Age	Age of patient
4	Weight	Weight of patient
5	WBC Count	White Blood Corpuscles
6	Cancer cell Count	Cancer cell of patient
7	Cell size (Micrometer)	Cell size in mm
8	Lump (Millimeter)	Lump in mm

Table-4.1: Parameters details of Dataset

4.1.1 **Proposed System**

Our model consisting of six stages that are demonstrated in below block diagram & that are dataset preparation (agricultural data are collected from various sources to prepare data set), preprocessing of data, feature extraction, algorithm (RF, SVM, KNN), recommendation system.

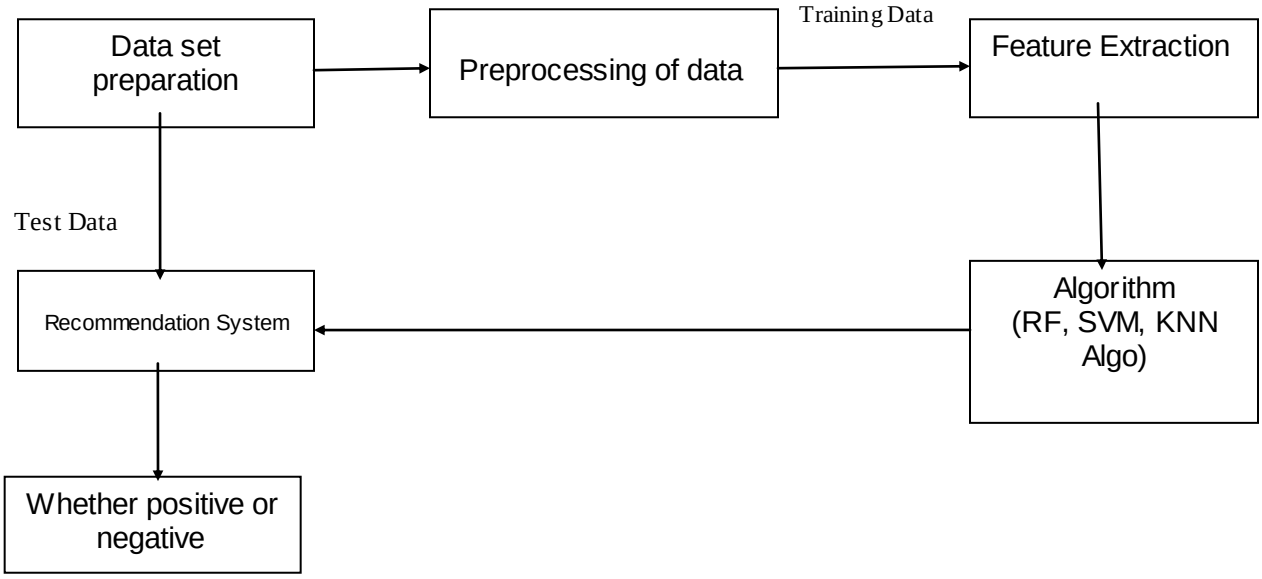

Fig 4.2: Block diagram of proposed framework

4.2 DATA PREPROCESSING

Prior information is important step to predict the blood cancer. It includes cleaning, transforming and reducing data to make it suitable for analysis.

- Data cleaning involves removing or correcting errors or inconsistencies of the parameters from our dataset.
- This may remove duplicates, filling in missing data.

- Then Data conversion will transform the data into a more suitable format for analysis.

Data reduction involves reducing the volume of data to simplify management analytically. Feature engineering involves creating new features or changes from existing data that will be more relevant to the problem. This may include combining existing features or creating new features. Overall, data preprocessing building is an accurate and effective blood cancer detection.

4.3.1 Software Requirements:

1. Software:
   Thonny 4.1.2
   Python 3 or above
2. Operating System: Windows 11
3. Python Libraries: NumPy, pandas, sklearn, matplotlib, seaborn

Thonny

Thorny is an integrated development environment for Python. It supports a variety of methods, from actions to evaluation of ad-hoc instructions, a detailed view of the call group, and models that describe usage and stack content. The program is available for Windows, macOS and Linux.

Thonny is a Python IDE available for machine learning. It is a lightweight and user-friendly IDE that provides a framework for building, debugging and testing Python code. While Thonny is a suitable IDE for general Python development, it may workable on complex machine learning tasks.

However, Thonny is for simple machine learning tasks that do not require advanced features. Thonny can write and run code and provides a debugger, code completion, syntax highlighting, which are important features for development.

Python3:

Python3 is the latest version of Python. It is a successor to Python 2, and it introduced many new features and improvements language.

Machine Learning

Machine learning is an artificial intelligence application where a machine learns from from input data like patient ID , WBC Count , weight, cancer cell Count, cell size, body temperature, age. Machine learning can be classified in three categories. Supervised learning can be again classified in two categories like classification & regression. We are used here RF, KNN & SVM which are supervised learning i.e. works on labeled data.

ML is an application of AI allows systems to learn and improve automatically through experience without special programming by programmers. The main goal is to allow computers to correct behavior without human interference to increase accuracy and usefulness of the program. Traditional writing of computer programs can be defined as the automation of procedures to be performed on input data to produce output artifacts. It is almost always linear, procedural and logical. A typical program is written in a programming language for several specifications and has properties such as inputs of our program.

Supervised ML can be broadly classified into the following types: Supervised learning takes feature/label pairs called training sets. From this training set, the system generates a general model of the relationship between a set of descriptive features and a target feature in form of program containing set of rules. The goal is to use the generated output program to predict labels for previously unseen input features, that is, to predict the outcome of some new data.
Data with known features not included in training set are classified according to established models and results compared with known features. This data set is called a test set. The ML can solve complex problems faster, more accurately, and at greater scale than manual solutions. The tools usually for ML include libraries that implement ML algorithms.

NumPy

NumPy is an open-source numerical Python library to compute scientifically. NumPy array is a powerful N-dimensional array object which is in the form of rows and columns. We can initialize NumPy arrays from nested Python lists and access it elements.

Pandas

A Panda is a library used for working with our dataset & it can analyze the high data structure, statistical data. Pandas allow us to analyze datasets and make decisions based on statistical datasets.

It's similar to a Python list or NumPy array, but adds the ability to tag each item, making easier for using.

Pandas provide a range of functions for manipulating data, including filtering, grouping, sorting, and merging & its tools used for cleaning and transforming messy data, such as filling missing values. removing duplicates. Pandas uses for data visualization using its built-in plotting functions or by integrating with other plotting libraries as Matplotlib and Seaborn. It also provides support for working with time series data, including functions for re-sampling, shifting, and rolling.

Sk-Learn

Scikit-learn (Sklearn) is the python library used for statistical modeling. This can work on both unsupervised and supervised learning algorithms.

The properties of scikit-learn provides include:

- Test data sets & training data sets with specific properties
- To reduce dimension
- Combines multiple supervised methods.
- Feature extraction & feature selection
- To make a model

Matplotlib

Matplotlib library works on two dimensional arrays. It is a visualization library on numpy. It can also visualize a huge data in several plots. Matplotlib pyplot makes matplotlib using collected functions. Matplotlib is an extension of Numpy.

Matplotlib provides many charts to choose from, including charts, graphs, charts, histograms, charts, and more. It allows extensive customization, including control of axes, names, colors, fonts, and more & provides tools for creating 3D drawings, including surface plots, pitch lines, and contour plots. Matplotlib integrates with other Python libraries, including NumPy and Pandas, to easily view data from these sources & used for computational science, data visualization, machine learning, etc. It is widely used in fields and Python scientific computing ecosystem.

Histogram is the distribution of numerical data. Histogram uses to display continuous data from our dataset in categorical form.

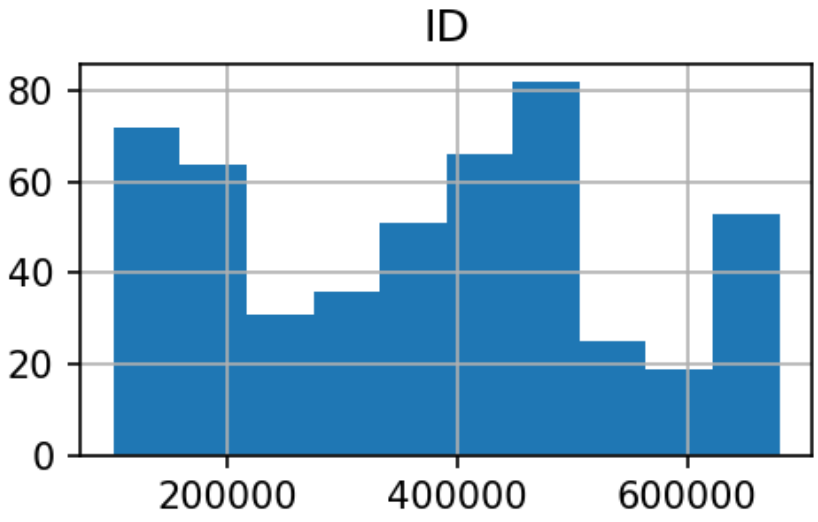

Fig-4.3: Histogram presentation of parameter ID

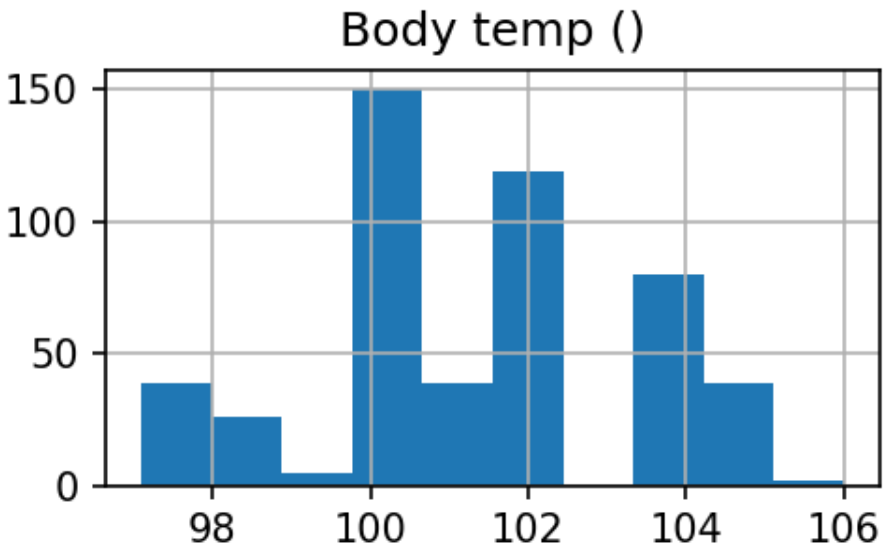

Fig 4.4: Histogram presentation of parameter Body temperature

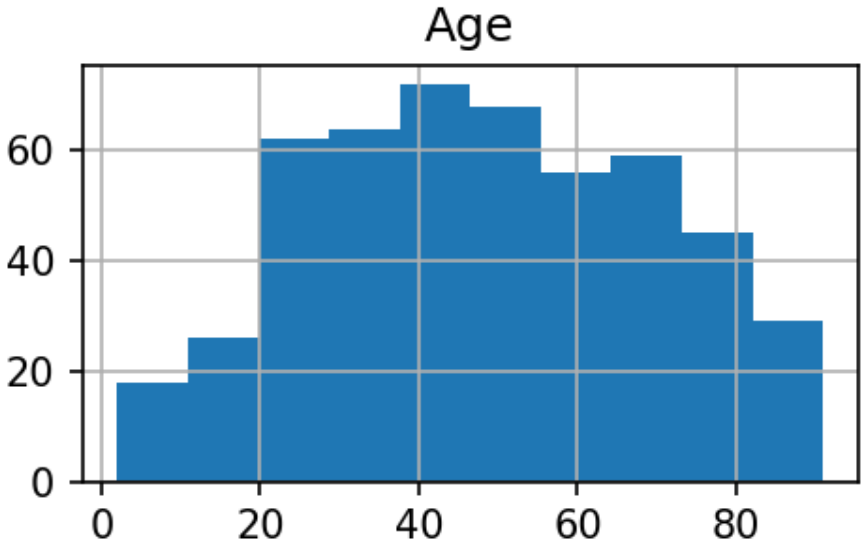

Fig 4.5: Histogram presentation of parameter Age

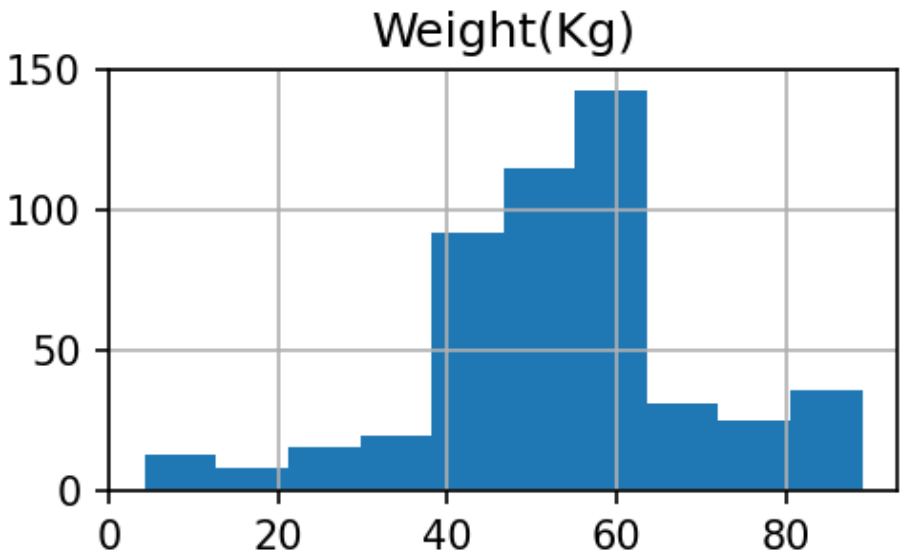

Fig 4.6: Histogram presentation of parameter weight

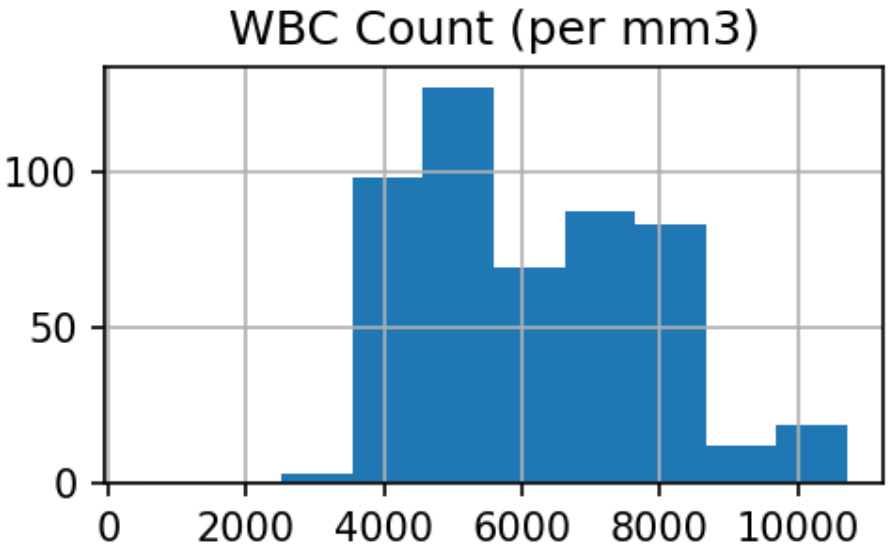

Fig 4.7: Histogram presentation of parameter WBC Count

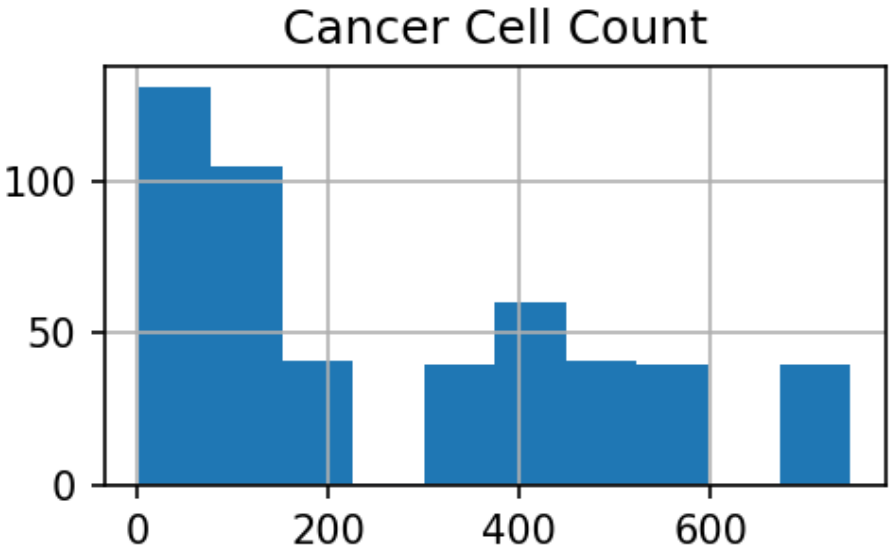

Fig 4.8: Histogram presentation of parameter Cancer cell Count

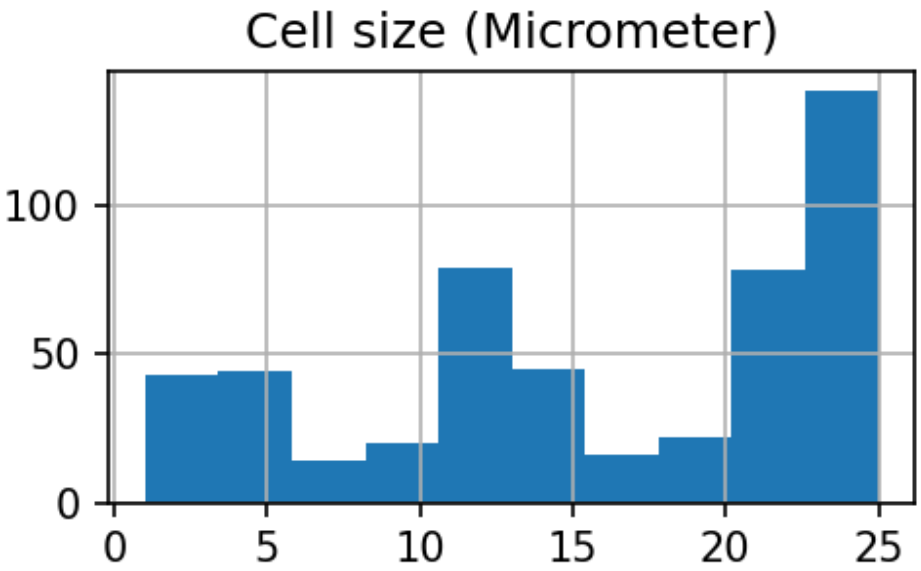

Fig 4.9: Histogram presentation of parameter cell size

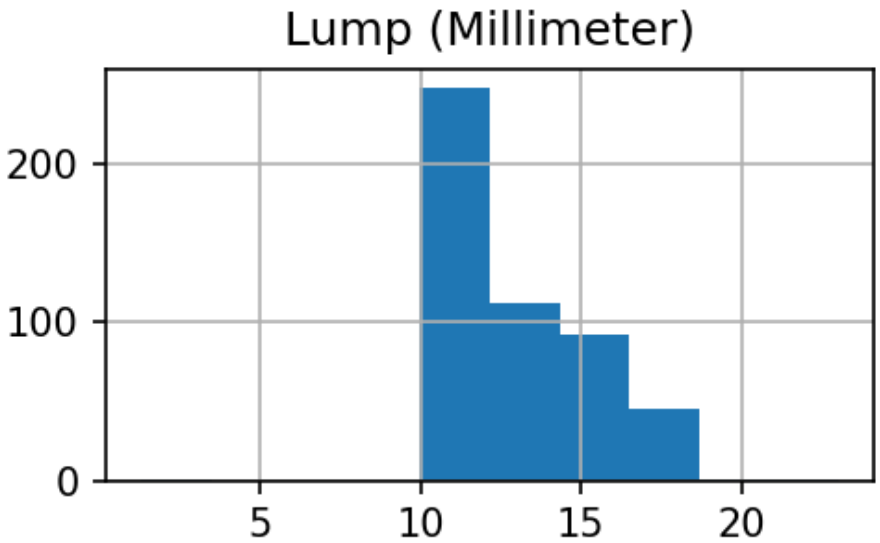

Fig 4.10: Histogram presentation of parameter lump

The main purpose of using a scatter chart is to analyze and show the relationship of two different numbers. The content in the scatter map is not just the value of a direction, but the pattern of the data as a whole. Analysis of relations is available in the scatter charts. The relationship may be positive, negative, strong ,weak, linear, non linear.

Seaborn

Seaborn uses for statistical plotting graphically presented in Python and to create statistical plot. The features of seaborn are

- Plotting statistical time series data
- Works with NumPy and Pandas data structures
- It creates Matplotlib graphics

1. Statistical Visualization: Seaborn provides a set of statistical visualizations designed to explore and understand complex data. These include scatter charts, line charts, bar charts, heat maps, and more.
2. Data exploration tools: Seaborn provides tools for data exploration, including ability of visualize distributions, identify outliers, and analyze trends, relationship between differences.
3. Customization: Seaborn allows a high level of customization, including control of color palettes, layouts and visual elements.
4. Integration: Easily integrates with other Python libraries, including Seaborn, Pandas, and Matplotlib.

4.3.2 Hardware requirements:

1. CPU: Multi-core CPUs can increase the training of large samples and provide parallelism of many data points.
2. RAM: Large files and complex models may require a lot of RAM (Random Access

Memory) to perform well. 16GB or more of RAM is recommended for most machine learning tasks.

4. Storage: Large files require huge storage space and read/write speed is vital for efficient operation.

Hardware requirements for ML may vary depending on specific tasks and data being processed.

In some cases, even a medium-sized computer will suffice for small-scale machine learning tasks, while larger datasets and more complex models can require significant resources.

4.3 .3 Splitting the Dataset into the Training set and Test set:

This training data is feed into the system to teach the machine how it works. Similar to humans, AI training data is similar to the instructions and experiences we enCounter as we learn, process data, and make decisions.

Because ML data is essential for prediction and efficiency needs to be under human control from beginning to end of process training. If an algorithm is well trained, it successfully identify features, make connections with data, and increase performance.

A sample of data from a training model that is often used the prediction of model's ability when adjusting the model's hyper parameters.

Actual test data are different from test data that are not included in training model but used to provide an unbiased assessment of the skills of the final adjusted model for selection or sample comparison.

For fit the model training data is used. The proof of validation is data set used to provide an objective assessment of how model fits the data subject to the hyper parameters. Evaluations are increasing ability to analyze data is important part of setting a standard.

Test data is used for evaluation. There are other ways to calculate biased estimates or to increase model skill estimates on data that is not visible in context of current data.

In ML data preprocessing, we divide our dataset into two sets. This is very crucial steps of data pre-processing as by doing this as we can enhance the performances of our ML model. The training dataset is used to train the model & develop the model. Test data set can use after training is done. Training data can differ on center of supervised & unsupervised algorithms.

Once our machine learning model has been trained 80%, 70%, 60%, 50% of data from our dataset as training data & 20%, 30%, 40%, 50% of data as test data respectively. In this phase we also can check the accuracy of our proposed system.

4.4 CLASSIFICATION MODEL BUILDING

It is analytical method for estimating the current level given categorical data. To create our model RF, SVM, KNN classifiers are used.

4.4.1 Random Forest Classifier

RF classifiers can be useful tools in the recommendation process because they can help farmers decide which crops to plant in which area, especially based on the unique environment of blood cancer.

Detection of blood cancer is on basis of seven parameters like patient IDs, body temperature, weight records, White Blood Cell (WBC) Counts, cancer cell Counts, cell sizes, and size of lumps makes informed decisions in advance to detect the blood cancer in early. RF algorithm selects random sample from training dataset. Then it will construct decision tree. After that each result is predicted considering voting. Final prediction will be done based on most voting result.

```
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=400)
classifier = RandomForestClassifier(n_estimators=100)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
a=accuracy_score(y_test, y_pred)
print("Random Forest Model Accuracy is:",a )
```

Fig-4.11: Implementation of Random Forest

Test Data	Training Data	Accuracy
100	400	0.97
150	350	0.98
200	300	0.97
250	250	0.94

Table-4.2: Accuracy table of Random Forest

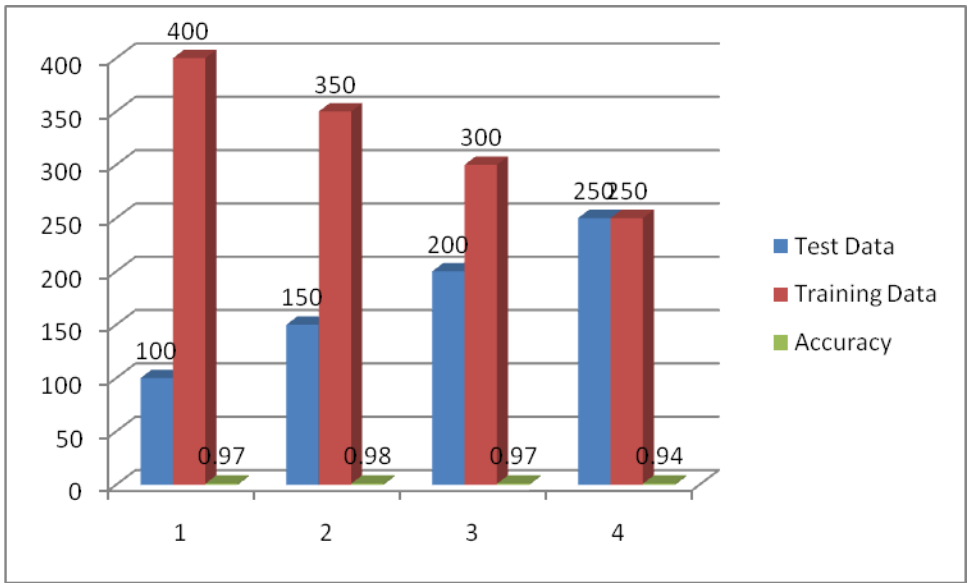

Fig-4.12: Bar chart test data, training data & accuracy of RF

Percentage of test data & train data are taken from our dataset as 20%, 30%,40%,50%,60% test data and remaining 80%, 70%,60%,50%,40% training data respectively.

4.4.2 SVM

SVM is concerned with binary data classification. Basis on seven parameters, SVMs determines whether the patient is blood cancer affected or not. This method creates a large plane in N-dimensional space, where N is the number of features that will be used to describe the content of the data in the dataset. The greater the distance of two points, the more accurate the distribution.

```
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.3, random_state=350)
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
svc = SVC(C=0.3, random_state=350, kernel='linear')
svc.fit(X_train_std, y_train)
y_predict = svc.predict(X_test_std)
b=accuracy_score(y_test, y_predict)
print("SVM Accuracy score  is ",b)
```

Fig- 4.13: Implementation of SVM

Test Data	Training Data	Accuracy
100	400	0.98
150	350	0.95
200	300	0.95
250	250	0.96

Table-4.3: Accuracy table of SVM

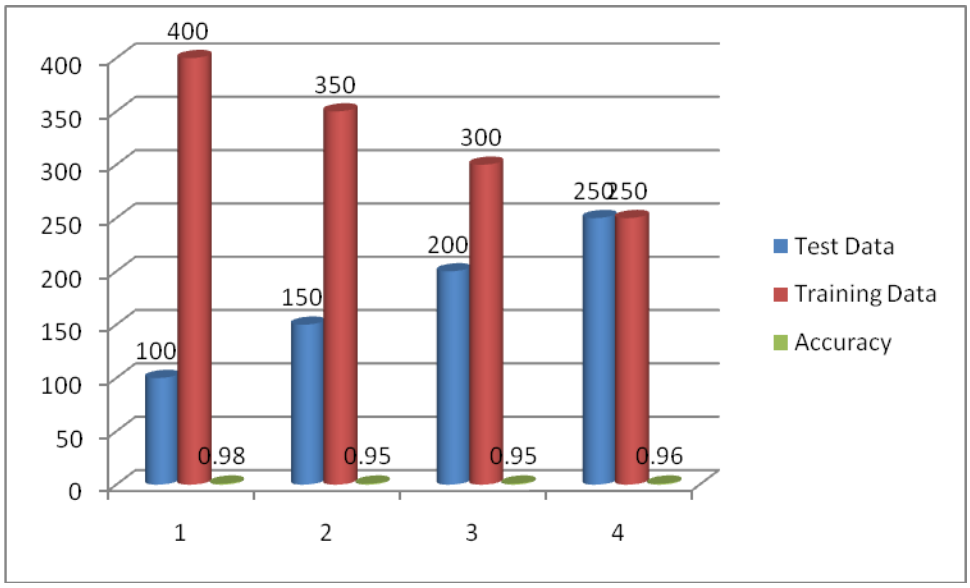

Fig-4.14: Bar chart test data, training data & accuracy of SVM

4.4.3 KNN

The k-Nearest Neighbors (KNN) algorithm is a valuable tool for blood cancer detection, especially when considering the distinct environmental factors that influence the occurrence of this disease. In the realm of agriculture, KNN, similar to RF classifiers, assists in decision-making for crop selection based on the specific environmental conditions of a given area.In the context of blood cancer detection, KNN operates on the premise of utilizing patient parameters to make informed decisions for early detection. These parameters include patient IDs, body temperature, weight records, White Blood Cell (WBC) Counts, cancer cell Counts, cell sizes, and size of lumps.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4, random_state=300)
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(X_train_std, y_train)
y_pred = knn.predict(X_test_std)
accuracy = accuracy_score(y_test, y_pred)
print("KNN Model Accuracy:", accuracy)
```

Fig- 4.15: Implementation of KNN

Test Data	Training Data	Accuracy
100	400	0.93
150	350	0.91
200	300	0.91
250	250	0.90

Table-4.4: Accuracy table of KNN

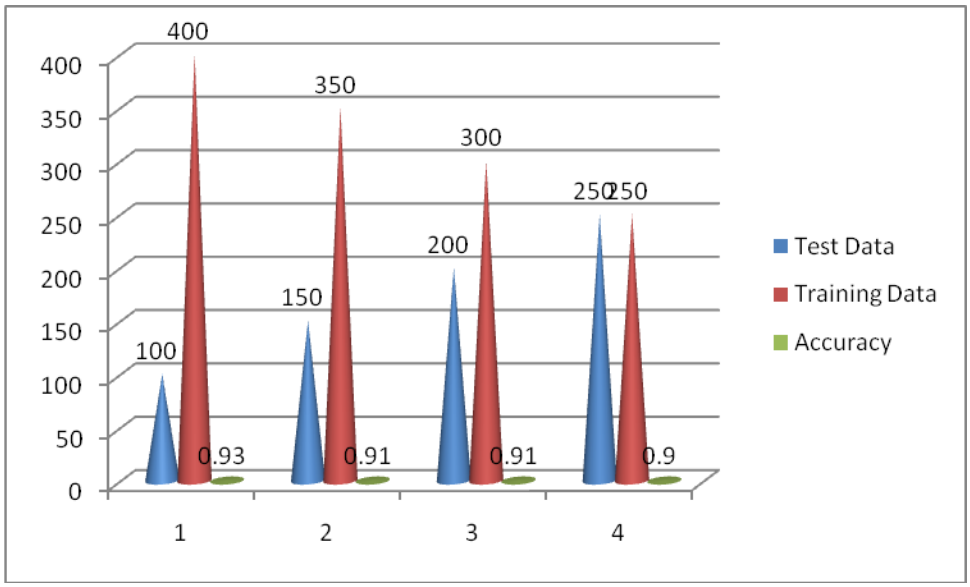

Fig-4.16 Bar chart test data, training data & accuracy of KNN

5 MODEL EVALUATION

Accuracy level in RF is better than SVM & KNN. So, RF shows better accuracy than other algorithm in our proposed system.

RF	SVM	KNN
0.97	0.98	0.93
0.98	0.95	0.91
0.97	0.95	0.91
0.94	0.96	0.90

Table 5: Comparison of accuracy table

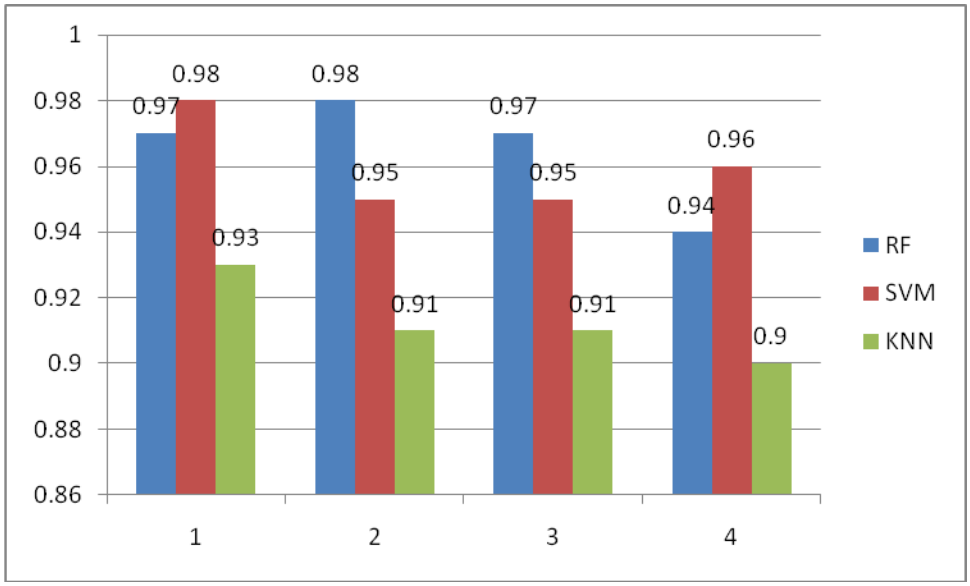

Fig-5 Evaluation of accuracy

RF	SVM	KNN
0.97	0.98	0.93
0.98	0.95	0.91
0.97	0.95	0.91
0.94	0.96	0.9
0.965	0.96	0.9125

Table 5: Comparison of average accuracy

Input Parameter	Value
ID	140221
Body temp	99.9
Age	55
Weight	59.3
WBC Count	4000
Cancer cell Count	534
Cell size	22.1
Lump size	15.3

Table 5.1: Input data for Cancer detection

Output:

```
['Positive']
        ID  Body temp ()  ...  Cell size (Micrometer)  Lump (Millimeter)
47  140223         102.4  ...                     2.3               10.2

[1 rows x 8 columns]
```

Fig- 5.1: Detection of the blood cancer using ensemble RF

```
X=mydata.iloc[:,0:-1]
mydata= mydata.iloc[: , :-1]
mydata=np.array([[400135,98.8,42,51.5,4500,123,13.21,11.2]])
prediction=RF.predict(mydata)
print(prediction)
```

output:

```
['Positive']
>>>
```

Fig- 5.2: Detection of the blood cancer using entered parameter

6. CONCLUSION AND FUTURE WORK.

The project focused on enhancing early detection of blood cancer has marked significant progress in leveraging machine learning algorithms—Support Vector Machines (SVM), Random Forest, and k-Nearest Neighbors (KNN). These algorithms were employed to predict blood cancer status based on patient parameters like body temperature, weight records, cell Counts, and other relevant factors.

Throughout the project, these algorithms showcased promising accuracy in predicting blood cancer status. This suggests their potential in early detection and intervention, which could significantly improve patient outcomes. Moreover, insights gained from the project shed light on crucial parameters that significantly influence blood cancer predictions. This knowledge aids healthcare professionals in identifying high-risk individuals and initiating proactive measures.
The validation and evaluation of SVM, Random Forest, and KNN models validated their

efficacy in blood cancer detection. This lays the foundation for their potential integration into clinical practice for early screening.

Future work in this domain could refine and optimize machine learning algorithms to enhance predictive accuracy further. Collaborating with healthcare institutions for real-time integration of predictive models into clinical workflows could facilitate early screening programs and risk stratification for blood cancer.

Incorporating additional datasets or parameters, such as genetic markers or specific blood cell characteristics, could enrich the model's predictive capabilities. Emphasis on ethical guidelines, patient consent, and regulatory compliance ensures responsible deployment of predictive models in healthcare settings, safeguarding patient privacy and trust.
Longitudinal studies to track patient outcomes and assess the actual impact of early detection on treatment effectiveness would provide valuable insights. Continued efforts in public health campaigns and healthcare professional training could foster a culture of proactive healthcare seeking behavior and improve patient outcomes.

Ultimately, this project lays the groundwork for leveraging machine learning algorithms in early detection efforts for blood cancer. Future endeavors in this direction have the potential to substantially advance healthcare practices, benefiting individuals at risk of blood cancer and contributing to better public health outcomes. We are trying to make app development using the said parameter to predict the early stages of blood cancer.

7. RELATED WORK

Saranyan, et al. in their 2021 paper, presented that Machine learning methods are currently being used in medical fields for identification and classification of blood cancer disease in early stages . Leukemia is a blood cancer caused by abnormal White Blood Cell (WBC) growth. Historically, diagnosing diseases and Counting blood cells required a lot of manual labor and instruments like the hemocytometer, which produced labor-intensive and inaccurate findings.

The authors suggested a computer-based alternative that makes use of medical image processing methods. This technique, which accurately Counts Red Blood Cells (RBCs), White Blood Cells (WBCs), and Platelets by examining microscopic images, makes it possible to identify suspected leukemia cases in addition to cell Counts. This development provides a more effective and precise approach to disease detection and cell Counting in compared to other manual methods.

8. REFERENCE

- Puneet , Anamika Chauhan. " Supervised detection of Lung cancer using Machine learning techniques based on routine blood indices" In 2020 IEEE International Conference for Innovation in Technology (INOCON)Bengaluru, India. Nov 6-8, 2020.
- R. Duggal, A. Gupta, and R. Gupta, "Segmentation of overlapping/touching white blood cell nuclei using artificial neural networks" CME Series on Hemato-Oncopathology. All India Institute of Medical Sciences (AIIMS), New Delhi, India, July 2016.
- S. Kumar, S. Mishra, P. Asthana," Automated Detection of AcuteLeukemia Using K-mean Clustering Algorithm". InAdvances inComputer and Computational Sciences. 2018, pp. 655-670
- Tomasz Markiewicz, Stanislaw Osowski, Bonenza Marianska, and Leszek Moszczynski. Automatic recognition of the blood cells of myelogenous leukemia using svm. In Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005., volume 4, pages 2496-2501.IEEE, 2005

APENDIX

Sample Code:

```
import pandas as pd
import numpy as np
import math
from sklearn import metrics
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import make_moons
from sklearn. import RandomForestClassifier
from sklearn. import VotingClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler, MinMaxScaler
from sklearn.svm import SVC

from matplotlib import pyplot
from pandas.plotting import scatter_matrix
mydata = pd.read_csv("C:\cancer_n1.csv")
mydata.info()
print(mydata)
mydata.hist()
pyplot.show()

X=mydata.iloc[:,0:-1]
print(X)
y=mydata.iloc[:,-1]
print(y)
```

```
X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=400)
sc = StandardScaler()
sc.fit(X_train)
X_train_std = sc.transform(X_train)
X_test_std = sc.transform(X_test)
svc = SVC(C=0.3, random_state=350, kernel='linear')
svc.fit(X_train_std, y_train)
y_predict = svc.predict(X_test_std)
b=accuracy_score(y_test, y_predict)
print("SVM Accuracy score  is ",b)

X_train, X_test, y_train, y_test = train_test_split(
X, y, test_size=0.2, random_state=400)
classifier = RandomForestClassifier(n_estimators=100)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
a=accuracy_score(y_test, y_pred)
print("Random Forest Model Accuracy is:",a )
```

Plagiarism Checker X - Report

Originality Assessment

13%

Overall Similarity

Date: Nov 26, 2023
Matches: 802 / 6070 words
Sources: 32

Remarks: Low similarity detected, check with your supervisor if changes are required.

Verify Report:
Scan this QR Code