IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

OVARIAN CANCER DETECTION USING MACHINE LEARNING ALGORITHMS

[1]Khushbu Leuva, [2]Hiteshkumar Parmar, [3]Janki Patel

[1]Assistant Professor, [2] Assistant Professor, [3]Lecturer

[1]Computer Engineering,

[1]Hasmukh Goswami College of engineering, Vahelal, Gujarat, India

ABSTRACT: Ovarian cancer remains a strong foe in the arena of women's health, ranking as one of the major causes of cancer-related death, especially when not discovered early. The current diagnostic landscape is heavily reliant on a multifaceted approach involving surgical interventions, ancestral lineage assessments, imaging techniques such as ultrasound and CT-Scans, and specialized blood tests such as CA125, all of which aim to differentiate between benign and malignant ovarian tumors. Early identification of ovarian cancer is critical, and developing machine learning tools provide potential prospects. The ability of machine learning to comprehend complicated data and provide accurate forecasts has the potential to revolutionize the diagnosis and therapy of ovarian cancer. Notably, numerous machine learning algorithms, like as Naive Bayes and Simple Regression, have demonstrate and their diagnostic capability in the diagnosis of ovarian cancer, with accuracies of 89.25% and 88.17%, respectively, across multiple repositories. The continuing study's major goal is to investigate and use various machine learning algorithms in the detection of ovarian cancer. This study aims to demonstrate a wide range of machine learning approaches designed for the early and accurate detection of both malignant and benign tumors linked with ovarian cancer. The investigation seeks not only to improve diagnosis accuracy, but also to shorten the procedure, potentially improving the efficacy of early interventions and personalized treatment paths in the field of ovarian cancer management.

Keywords: **machine Learning, Ovarian Cancer detection.**

I. INTRODUCTION

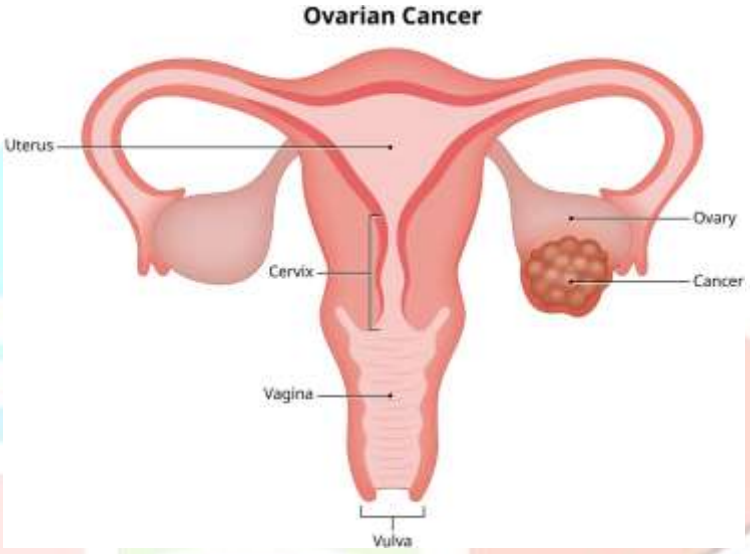
Cancer is a disease that happens when the body's aberrant cells multiply uncontrollably. Cancer is sometimes called for the portion of the body where it first appears, regardless of whether it eventually spreads to other regions of the body. Ovarian cancer is a category of disorders that can develop in the ovaries or in nearby organs such as the Fallopian tubes and peritoneum. In women, one ovary is positioned in the pelvis on each side of the uterus.

The ovaries are in charge of creating the female hormones and eggs needed for re- production. The Fallopian tubes are a pair of long, thin tubes found on either side of the uterus in women. The Fallopian tubes carry eggs from the ovaries to the uterus. The medical word for the tissue lining that protects the abdominal organs is the peritoneum. Early detection of ovarian cancer is critical for effective therapy. Ovarian cancer generally manifests as signs and symptoms, so pay attention to your

body and discover what's normal for you.

Consult your doctor, nurse, or other healthcare practitioner to determine whether your symptoms are caused by something other than cancer. There are several ovarian cancer tumor types and sub types. Adenocarcinoma is the most common cancer kind, and serous adenocarcinoma is the most common sub type of this tumours type. Serous adenocarcinoma is a kind of high-grade, rapidly developing tumour [1] only females (women) have ovaries, or female reproductive glands. For reproduction, the ovaries produce eggs (ova). The fertilized egg enters the uterus after moving from the ovaries through the Fallopian tubes and developing into a fetus.

The ovaries also produce the feminine hormones progesterone and estrogen. Each side of the uterus has one ovary. The bulk of the cells in the ovaries are divided into three categories. Each cell type has the potential to generate a different type of tumours epithelial tumours form from cells that line the exterior of the ovary. Epithelial cell tumours account for the vast majority of ovarian



malignancies. Ovarian cells, or ova, are used to make germ cell tumours. Stromal tumours develop from the cells that keep the ovary together and generate the female hormones progesterone and estrogen.

figure 1.1: ovarian cancer [1]

Atypically developing ovarian cells grow into tumours [2] which may or may not be malignant. If the malignant cells or tumours were not detected in time, they may have spread to the pelvic region, ovary, stomach, and other organs. When developing a treatment plan, the stage of the ovarian cancer must be carefully considered.

.

II. RELATED WORK:

Khalsan et al [8] identify the paper "A Survey of Machine Learning Approaches Applied to Gene Expression Analysis for Cancer Prediction" likely delves into the fusion of machine learning and genomics within the cancer research domain. It addresses the important function of gene expression patterns in predicting distinct cancer types and the application of machine learning methods to assess these genetic markers. This survey may detail the diverse landscape of machine learning methodologies utilized in cancer prediction, ranging from supervised methods like decision trees, support vector machines, and neural networks to unsupervised techniques such as clustering algorithms. The examination of issues in data selection, feature extraction, and the interpretability of machine learning models applied to gene expression analysis for cancer prediction are central to the article. Moreover, it might highlight successful applications of these models, emphasizing their

potential in advancing early detection and personalized treatment strategies in oncology.

Sai Jahanavi et al [9] identify the paper "Predicting Ovarian Cancer using Machine Learning" delves into the application of various machine learning techniques for the early detection and prognosis of ovarian cancer. It describes how to analyses datasets comprising a variety of patient-related characteristics, genetic markers, and clinical indicators using various algorithms such as Naive Bayes, Simple Regression, Support Vector Machines, or ensemble approaches. The project intends to construct prediction models capable of recognizing patterns or signs suggestive of ovarian cancer at an early stage by combining large data and algorithmic analysis. These models not only aid in accurate diagnosis, but also in anticipating potential outcomes and responsiveness to therapies, strengthening personalized care, and improving overall ovarian cancer patient management. The paper's major focus is on harnessing the power of machine learning to develop powerful prediction tools that help in early identification and better informed decision-making in the diagnosis and treatment of ovarian cancer.

Liberto et al [10] identify the paper "Current and emerging methods for ovarian cancer screening and diagnostics: a comprehensive review," delves into the wide spectrum of strategies and techniques avail- able for the early detection and diagnosis of ovarian cancer. It will most likely investigate known methods such as imaging technology (ultrasound, CT scans), blood biomarkers (CA125), and genetic testing, as well as the promising landscape of developing options. These might include advances in molecular diagnostics such as circulating tumours DNA, proteomic analysis, and the use of artificial intelligence in diagnostic data interpretation. The comprehensive review will most likely evaluate each method's strengths, limitations, and potential future implications, with the goal of providing a detailed overview of the current state and future prospects in ovarian cancer screening and diagnostics, potentially contributing valuable insights to ongoing efforts to improve early detection and patient outcomes in ovarian cancer management.

Wibowo et al [11] identify the paper titled "Ovarian Cancer Classification using K-Nearest Neighbor and Support Vector Machine" delves into the application of machine learning algorithms—specifically, the K-Nearest Neighbor (K-NN) and Support Vector Machine (SVM) for the classification of ovarian cancer. The algorithms will most likely be used to analyses datasets comprising various traits or variables related with ovarian cancer patients in the research. K-NN classifies instances based on the majority of their neighboring instances, whereas SVM seeks to separate data into multiple classes by locating the best hyper plane. The research will most likely investigate the usefulness of these machine learning algorithms in properly identifying ovarian cancer cases, possibly addressing concerns such as early diagnosis, differentiating benign from malignant tumours, and optimizing treatment choices. The study's goal could be to compare the performance, accuracy, and comparative advantages of K-NN and SVM in the context of ovarian cancer classification, potentially providing valuable insights to improve diagnostic methodologies in oncology.

Jaiswal et al [12] identify the paper review on Machine Learning Algorithm in Cancer Prognosis and Prediction" comprehensively explores the application of machine learning in the realm of cancer prognosis and prediction. .The study investigates the usefulness of several machine learning algorithms in assisting the prognosis and prediction of various forms of cancer by methodically analyzing a plethora of publications and research initiatives. It provides a comprehensive review of these algorithms' strengths and weaknesses in understanding complicated medical data spanning from genetic markers and imaging results to clinical records. The study highlights the accuracy and efficiency of algorithms such as support vector machines, neural networks, decision trees, and ensemble approaches in predicting outcomes, recurrence, and treatment responses for various cancer

types. Furthermore, the study discusses the limitations and future potential of employing machine learning approaches, giving useful insights for future developments in personalized cancer care and treatment.

Nuhi´c et al [13] identify the paper "Comparative Study on Different Classification Techniques for Ovarian Cancer Detection" digs into a thorough examination of several classification approaches targeted at improving ovarian cancer diagnosis. The study will most likely compare and evaluate the performance, accuracy, and usefulness of approaches such as Naive Bayes, Support Vector Machines (SVM), Decision Trees, Random Forest, and potentially others. The research seeks to analyses the strengths and limitations of each approach in effectively identifying ovarian cancer through thorough comparative analysis. Such research has great promise for improving diagnostic techniques, perhaps opening the way for more precise and efficient early detection measures, and eventually influencing the prognosis and treatment results of ovarian cancer patients.

Nuklianggraita et al [14] identify the paper titled "On the Feature Selection of Microarray Data for Cancer Detection based on Random Forest Classifier" delves into the crucial aspect of feature selection in cancer detection using microarray data. Specifically, It focuses on the use of a Random Forest Classifier, a strong machine learning method, to find the most significant characteristics from microarray data in order to enhance cancer diagnosis accuracy. Microarray technology allows for the simultaneous assessment of hundreds of genes' expression levels, allowing for the discovery of biomarkers associated with various forms of cancer. The work investigates the optimization of feature selection using the Random Forest Classifier, with the goal of identifying the most influential genetic markers that may efficiently classify between malignant and non-cancerous samples. The article most likely explores the usefulness of this strategy in improving cancer detection accuracy and may provide insights into possible biomarkers for different forms of cancer, leading to the evolution of more precise diagnostic tools and methods.

Arfiani et al [15] identify the paper It focuses on the use of a Random Forest Classifier, a strong machine learning method, to find the most significant characteristics from microarray data in order to enhance cancer diagnosis accuracy. Microarray technology allows for the simultaneous assessment of hundreds of genes' expression levels, allowing for the discovery of biomarkers associated with various forms of cancer. The work investigates the optimization of feature selection using the Random Forest Classifier, with the goal of identifying the most influential genetic markers that may efficiently classify between malignant and non-cancerous samples. The article most likely explores the usefulness of this strategy in improving cancer detection accuracy and may provide insights into possible biomarkers for different forms of cancer, leading to the evolution of more precise diagnostic tools and methods.

El-Bendary et al [16] identify the paper on "Epithelial ovarian cancer stage subtype classification using a clinical and gene expression integrative approach" delves into a sophisticated methodology aimed at classifying epithelial ovarian cancer (EOC) subtypes at various stages by integrating clinical data with gene expression profiles. This approach likely involves leveraging computational techniques to analyze both the genetic makeup and clinical information of EOC patients. By merging these diverse datasets, the study aims to create a more comprehensive understanding of the disease, potentially leading to a refined classification system for different EOC stages and subtypes. Such an integrated approach could significantly enhance our ability to tailor treatments and prognostic assessments for EOC, potentially paving the way for more personalized and effective therapeutic strategies based on the specific characteristics of the cancer sub types.

III. IMPLEMENTATION DETAILS OF PROPOSED WORK:

Data Merge of Ovarian cancer for Model:

Merging datasets for a machine learning model involves combining two or more datasets into a single data set that can be used for training, testing, or validating your model. This process is often necessary when you have multiple sources of data that are relevant to your problem.

Here are some steps you can follow to merge datasets for a machine learning model:

Understand Your Data: Before merging datasets, make sure you understand the structure and contents of each dataset. Identify common columns or features that can be used as keys for merging.

Data Cleaning: Ensure that the data in each dataset is clean and consistent. Handle missing values, outliers, and any other data issues.

Identify Common Columns: Identify the common columns or features that exist in both datasets. These will be used as keys for merging.

Choose the Merge Type: Decide on the type of merge operation you want to perform. Common types include inner, outer, left, and right merges. The choice depends on whether you want to keep only the common rows, all rows from one dataset, or all rows from both datasets.

Perform the Merge: Use a programming language like Python with libraries such as Pandas to perform the merge.

Handle Duplicates: Check for and handle any duplicate rows that may result from the merge.

Feature Engineering: After merging, you might want to engineer new features based on the combined data. This can involve creating new columns or transforming existing ones.

Data Splitting: If you are preparing data for machine learning, split the merged data set into training and testing sets. This is crucial to evaluate the performance of your model.

Validation: Validate the merged data set to ensure that it meets the requirements for your machine learning model.

Model Training: Use the merged data set to train your machine learning model.

figure 1.2: multiple data merge

Data Preparation of Ovarian cancer data set for models:

Data Loading:

Load the dataset into your preferred programming environment (e.g., Python with Pandas). Make sure to understand the structure of your dataset, including the features and the target variable.

```
In [1]: import pandas as pd
        import numpy as np

        pd.set_option('display.max_columns',None)

In [2]: data = pd.read_csv('ovarian_cancer_prediction_dataset.csv')
        data.head()
```

Out[2]:

	SUBJECT_ID	AFP	AG	Age	ALB	ALP	ALT	AST	BASO#	BASO%	BUN	Ca	CA125	CA19-9	CA72-4	CEA	CL	CO2CP	CREA	TYPE	DBIL	EO#	EO%	GGT	GLO
0	1	3.58(c)	19.36	47	45.4	56.0	11.0	24.0	0.01	0.30	5.35	2.48	15.36(c)	36.48(c)	6.42	1.40	107.4	19.9	103.0	0	2.0	0.04	1.00	16.0	28.5
1	2	34.24(c)	23.98	61	39.9	95.0	9.0	13.0	0.02	0.30	3.21	2.62	2444.00(c)	19.98(c)	NaN	2.46	100.1	22.1	45.0	0	2.6	0.04	0.50	13.0	32.1
2	3	1.50(c)	18.40	39	45.4	77.0	9.0	16.0	0.03	0.60	3.80	2.57	56.08(c)	12.18(c)	NaN	0.77	102.6	22.2	48.0	0	4.7	0.03	0.60	10.0	32.5
3	4	2.75	16.60	45	39.2	26.0	16.0	17.0	0.05	0.74	5.27	2.35	2555	18.41	131.60	0.82	103.2	24.0	65.7	0	2.9	0.00	0.07	17.0	26.9
4	5	2.36	19.97	45	35.0	47.0	21.0	27.0	0.01	0.10	4.09	2.48	1391	11.15	NaN	0.42	99.6	26.2	70.3	0	2.2	0.11	1.60	24.0	31.5

figure 1.3: data loading

Data Exploration:

Explore the dataset to understand its characteristics, check for missing values, and analyze the distribution of the target variable.

```
In [3]: ## Subject Id is just a serial number for dataset, removing it
        data = data.drop('SUBJECT_ID',axis=1)
        data.shape

Out[3]: (698, 50)

In [4]: # TYPE is Traget column, which categorical (0 = Ovarian Cancer, 1 Benign or No Cancer)
        # separating Target from data for data preprocessing
        X = data.drop('TYPE',axis=1)
        y = data.TYPE

In [5]: # Some columns (like AFP, CA125, CA19-9 contains '\t' in data, which is noise, removing it
        X.replace('\t', '', regex=True, inplace=True)

In [6]: # some data is (like '>1210.00' and '<0.006' in dataset, removing '>' and '<' sign and considering rest value, as it won't help identifying actual value.
        X.replace('>', '', regex=True, inplace=True)
        X.replace('<', '', regex=True, inplace=True)

In [7]: X.head()
```

Out[7]:

	AFP	AG	Age	ALB	ALP	ALT	AST	BASO#	BASO%	BUN	Ca	CA125	CA19-9	CA72-4	CEA	CL	CO2CP	CREA	DBIL	EO#	EO%	GGT	GLO	GLU.	HCT	HE4	HI
0	3.58	19.36	47	45.4	56.0	11.0	24.0	0.01	0.30	5.35	2.48	15.36	36.48	6.42	1.40	107.4	19.9	103.0	2.0	0.04	1.00	16.0	28.5	4.67	0.273	NaN	6
1	34.24	23.98	61	39.9	95.0	9.0	13.0	0.02	0.30	3.21	2.62	2444.00	19.98	NaN	2.46	100.1	22.1	45.0	2.6	0.04	0.50	13.0	32.1	10.50	0.417	934.10	12
2	1.50	18.40	39	45.4	77.0	9.0	16.0	0.03	0.60	3.80	2.57	56.08	12.18	NaN	0.77	102.6	22.2	48.0	4.7	0.03	0.60	10.0	32.5	4.64	0.391	47.56	13
3	2.75	16.60	45	39.2	26.0	16.0	17.0	0.05	0.74	5.27	2.35	2555	18.41	131.60	0.82	103.2	24.0	65.7	2.9	0.00	0.07	17.0	26.9	4.76	0.372	853.50	12
4	2.36	19.97	45	35.0	47.0	21.0	27.0	0.01	0.10	4.09	2.48	1391	11.15	NaN	0.42	99.6	26.2	70.3	2.2	0.11	1.60	24.0	31.5	4.07	0.383	404.90	12

figure 1.4: data exploration

Data Cleaning:

Handle missing values and outliers appropriately. This may involve imputing missing values or

```
In [8]: # checking datatypes of columns
        X.dtypes

Out[8]: AFP          object
        AG          float64
        Age           int64
        ALB         float64
        ALP         float64
        ALT         float64
        AST         float64
        BASO#       float64
        BASO%       float64
        BUN         float64
        Ca          float64
        CA125        object
        CA19-9       object
        CA72-4      float64
        CEA         float64
        CL          float64
        CO2CP       float64
        CREA        float64
        DBIL        float64
        EO#         float64
        EO%         float64
        GGT         float64
        GLO         float64
        GLU.        float64
        HCT         float64
        HE4         float64
        HGB         float64
        IBIL        float64
        K           float64
        LYM#        float64
```

removing rows/columns with a significant number of missing values.

figure 1.5: data cleaning

Feature Selection:

Select relevant features for your models. Remove any unnecessary or redundant features.

```
CA72-4      float64
CEA         float64
CL          float64
CO2CP       float64
CREA        float64
DBIL        float64
EO#         float64
EO%         float64
GGT         float64
GLO         float64
GLU.        float64
HCT         float64
HE4         float64
HGB         float64
IBIL        float64
K           float64
LYM#        float64
LYM%        float64
MCH         float64
MCV         float64
Menopause     int64
Mg          float64
MONO#       float64
MONO%       float64
MPV         float64
Na          float64
NEU         float64
PCT         float64
PDW         float64
PHOS        float64
PLT           int64
RBC         float64
RDW         float64
TBIL        float64
TP          float64
```

figure 1.6: features selection

Data Splitting:

Split the dataset into training and testing sets. This allows you to train your models on one subset and evaluate their performance on another.

```
In [9]: # some columns have object type but having float data actually inside of column. changing object type to float
        object_columns = X.select_dtypes(include='object').columns
        X[object_columns] = X[object_columns].astype(float)

In [10]: # Null value Analysis
         X.isnull().sum()

Out[10]: AFP       44
         AG         2
         Age        0
         ALB       20
         ALP       20
         ALT       20
         AST       20
         BASO#      0
         BASO%      0
         BUN        0
         Ca         0
         CA125     34
         CA19-9    48
         CA72-4   400
         CEA       44
         CL         0
         CO2CP      2
         CREA       0
         DBIL      20
         EO#        0
         EO%        0
         GGT       20
         GLO       20
         GLU.       0
         HCT        0
         HE4       40
         HGB        0
```

figure 1.7: data splitting

Feature Scaling:

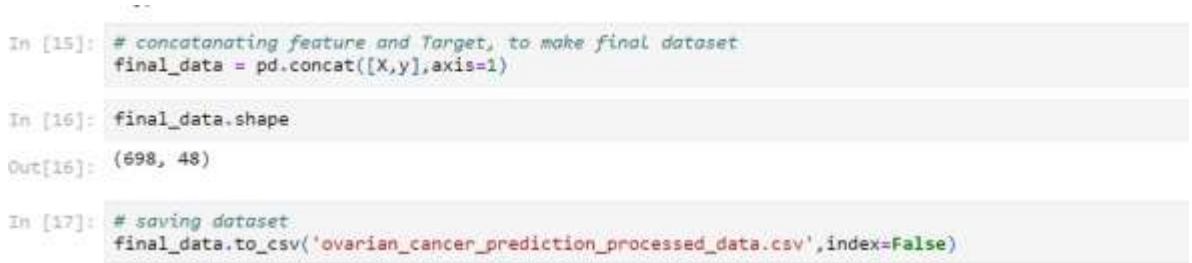
Scale the features if necessary, especially if you're using algorithms sensitive to the scale of input features.

```
In [11]: # NEU and CA72-4 have alot of missing values, hence this columns won't help in prediction, simply removing it.
         X = X.drop(['CA72-4','NEU'],axis=1)

In [12]: X.shape

Out[12]: (698, 47)

In [13]: # all columns which has null values are numeric, can replace null values with mean of the column
         X.fillna(X.mean(), inplace=True)

In [14]: # checking if is there any null values in column remaining or not.
         X.isnull().sum()

Out[14]: AFP       0
         AG        0
         Age       0
         ALB       0
         ALP       0
         ALT       0
         AST       0
         BASO#     0
         BASO%     0
         BUN       0
         Ca        0
         CA125     0
         CA19-9    0
         CEA       0
         CL        0
         CO2CP     0
         CREA      0
         DBIL      0
         EO#       0
         EO%       0
         GGT       0
         GLO       0
```

figure 1.8: feature scaling

Model Training:

Train each model using the prepared dataset.



```
In [15]: # concatanating feature and Target, to make final dataset
         final_data = pd.concat([X,y],axis=1)

In [16]: final_data.shape

Out[16]: (698, 48)

In [17]: # saving dataset
         final_data.to_csv('ovarian_cancer_prediction_processed_data.csv',index=False)
```

figure 1.9: train dataset

Implementation of Algorithms with Prepared Dataset:

Load the Dataset: Start by loading your ovarian cancer data set. You can use libraries like pandas to handle the data set.

Data Preprocessing: Handle missing values, encode categorical variables, and split the data into features (X) and target variable (y).

Split the Data into Training and Testing Sets: Split your data set into a training set and a testing set.

Build and Train Models: Now, build and train models using Decision Tree, SVM, Random Forest, and XGBoost.

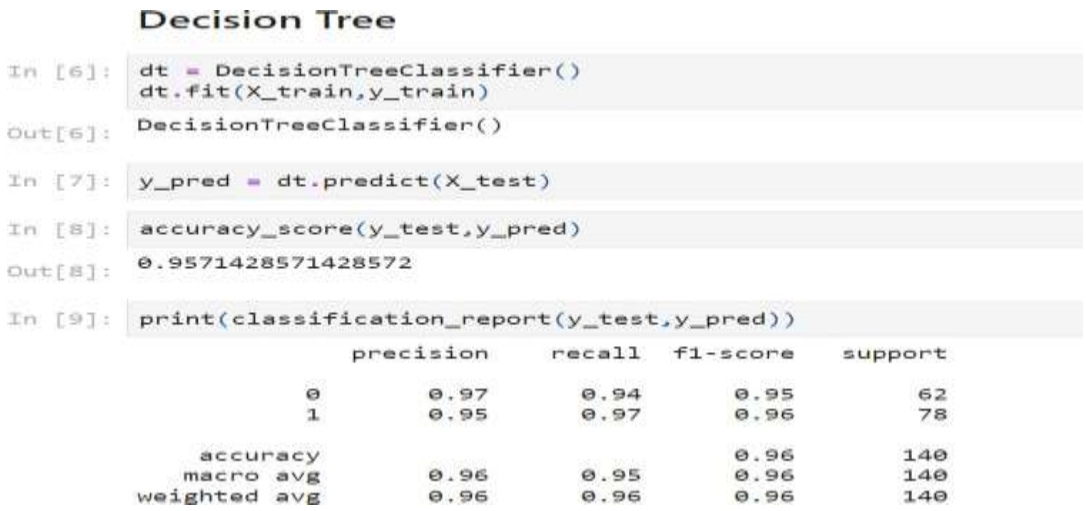
Evaluate Models: Evaluate the performance of each model on the testing set using metrics such as accuracy, precision, recall, and F1-score

```
In [1]: import pandas as pd
        import numpy as np
        import pickle
        from sklearn.model_selection import train_test_split, RandomizedSearchCV
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.ensemble import RandomForestClassifier
        from xgboost import XGBClassifier
        from sklearn.svm import SVC
        from sklearn.metrics import accuracy_score,f1_score,recall_score,classification_report,confusion_matrix, roc_curve, roc_auc_score
        import matplotlib.pyplot as plt
        import seaborn as sns

In [2]: data = pd.read_csv('ovarian_cancer_prediction_processed_data.csv')
        data.shape

Out[2]: (698, 48)

In [3]: ## removing target column from data
        X = data.drop('TYPE',axis=1)
        y = data.TYPE

In [4]: # dividing data into train test split
        X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=123)

In [5]: X_train.shape,y_train.shape

Out[5]: ((558, 47), (558,))
```

figure 1.10: prepared data set

Decision Tree Algorithm

The decision tree is a supervised machine learning approach that creates a tree-like structure with core nodes representing decision points and leaf nodes representing outcomes. It works by recursively partitioning data based on the most useful attributes. To find the best splits at each node, the algorithm chooses features and splitting criteria, such as mean squared error for regression and Gini impurity for classification. Until a halting condition is satisfied, such reaching a maximum tree depth or a minimum amount of samples in a leaf, this procedure keeps going backwards. Pruning strategies are frequently used to increase generality in decision trees since they are prone to over fitting. Decision trees may be easily implemented for a variety of applications using the Python scikit-learn module, which is suitable for both novice and expert users.



Decision Tree

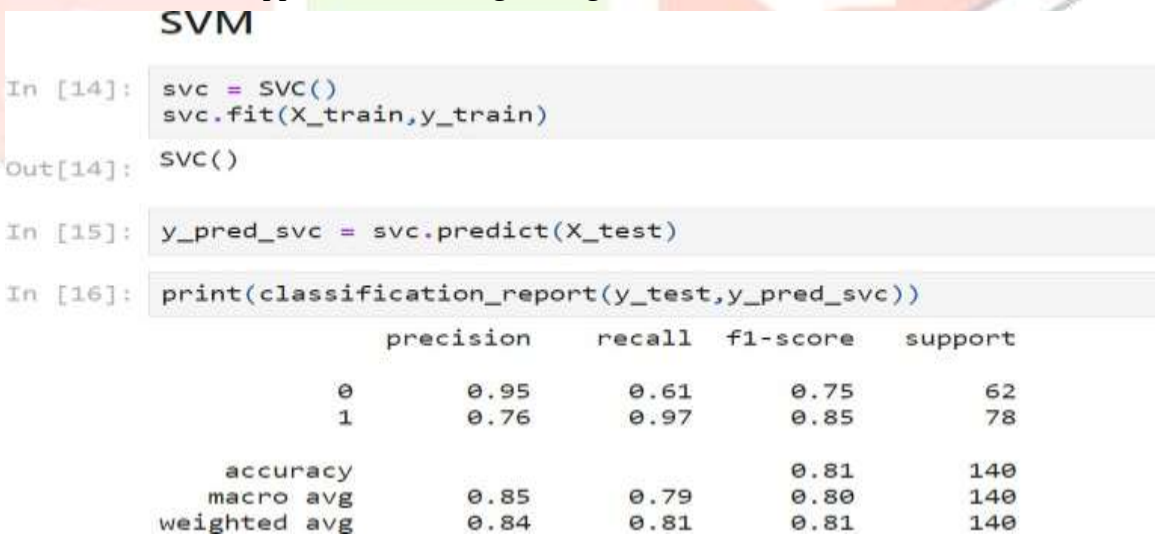
```
In [6]: dt = DecisionTreeClassifier()
        dt.fit(X_train,y_train)
Out[6]: DecisionTreeClassifier()

In [7]: y_pred = dt.predict(X_test)

In [8]: accuracy_score(y_test,y_pred)
Out[8]: 0.9571428571428572

In [9]: print(classification_report(y_test,y_pred))
```

	precision	recall	f1-score	support
0	0.97	0.94	0.95	62
1	0.95	0.97	0.96	78
accuracy			0.96	140
macro avg	0.96	0.95	0.96	140
weighted avg	0.96	0.96	0.96	140

figure 1.11: decision tree algorithm

Support Vector Machine Algorithm

Support Vector Machine (SVM) is a supervised machine learning algorithm designed for classification or regression tasks. It seeks to find a hyper plane that maximizes the margin between different classes in a high-dimensional feature space. The margin is the distance between the hyper plane and the nearest data points from each class. SVM uses a mathematical optimization approach to determine the optimal hyper plane, and it can handle non-linear decision boundaries through the use of kernels. The regularization parameter C influences the trade-off between achieving a low training error and a low testing error, making SVMs adaptable to various scenarios. With their versatility and effectiveness in high-dimensional spaces, SVMs have found applications in image recognition, text classification, and bioinformatics.



SVM

```
In [14]: svc = SVC()
         svc.fit(X_train,y_train)
Out[14]: SVC()

In [15]: y_pred_svc = svc.predict(X_test)

In [16]: print(classification_report(y_test,y_pred_svc))
```

	precision	recall	f1-score	support
0	0.95	0.61	0.75	62
1	0.76	0.97	0.85	78
accuracy			0.81	140
macro avg	0.85	0.79	0.80	140
weighted avg	0.84	0.81	0.81	140

figure 1.12: support vector machine algorithm

Random Forest Algorithm

Random Forest is an ensemble learning algorithm that builds multiple Decision Trees by introducing randomness in the training process. It leverages bootstrapped sampling of the dataset and random feature selection at each node to create diverse and independent trees. For classification tasks, the final prediction is determined by a majority vote, while for regression tasks, it's based on averaging the predictions. This method of combining multiple trees enhances predictive accuracy and generalization, mitigating over fitting. Random Forest is known for its robustness, adaptability to various data types, and the ability to identify feature importance, making it widely used for both classification and regression tasks in machine

learning.

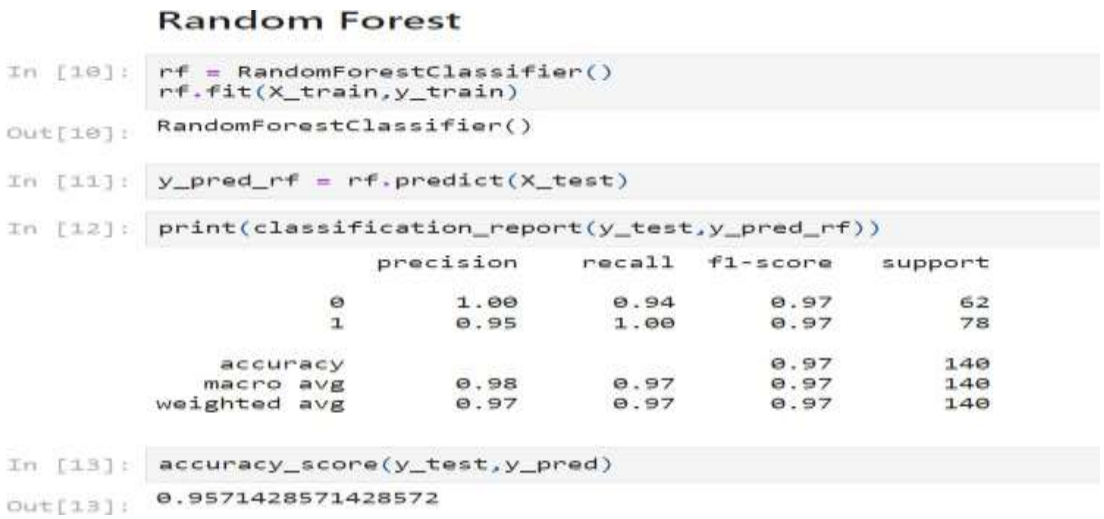


figure 1.13: random forest algorithm

XGBoost Algorithm

XGBoost, or extreme Gradient Boosting, is a highly effective and widely used machine learning algorithm that falls under the ensemble learning category. Leveraging gradient boosting, XGBoost sequentially builds a series of weak learners, often decision trees, and optimizes their combination to minimize a regularized objective function. Its key features include level-wise tree construction, gradient-based optimization for informed decision- making, regularization to prevent over fitting, efficient handling of missing values, and parallelization for scalability. XGBoost's success lies in its ability to deliver high predictive performance across diverse tasks, making it a popular choice in both competitions and real-world applications.

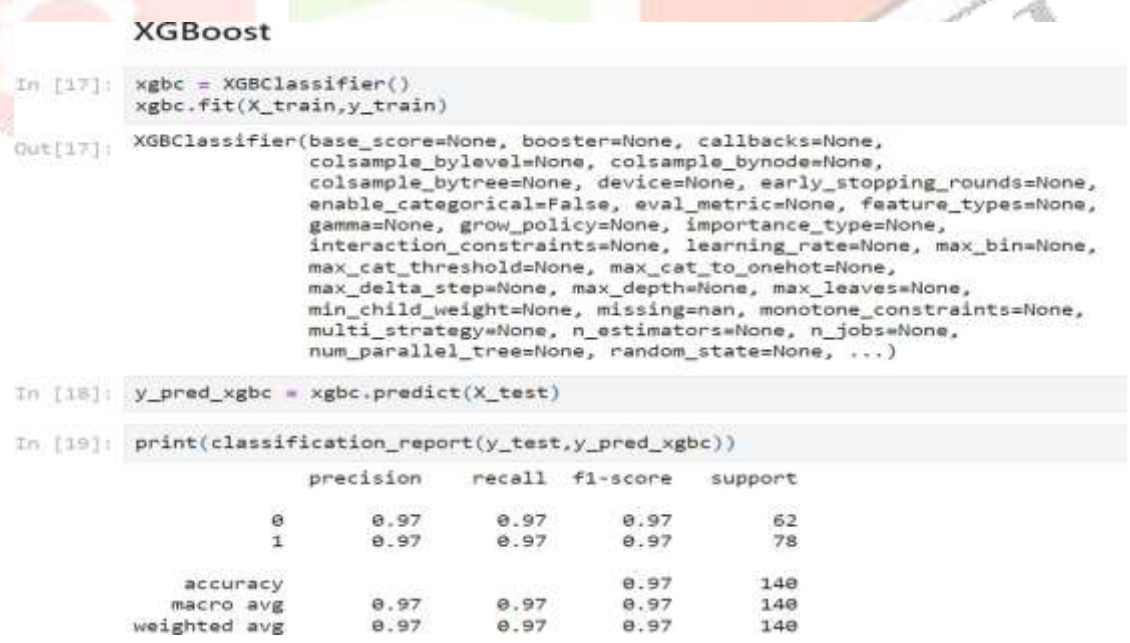


figure 1.14: XGBoost algorithm

III. RESULT ANALYSIS:

Data preprocessing involved several steps to prepare the ovarian cancer dataset for model training. Firstly, missing values were addressed through strategies like mean imputation or deletion of rows/columns, ensuring data integrity. Categorical variables were then en- coded using techniques such as one-hot encoding or label encoding, tailored to the specific data characteristics. Subsequently, the dataset was split into features (X) and the target variable (y) to facilitate model training.

Model implementation and training encompassed four distinct algorithms: Decision Tree, Support Vector Machine (SVM), Random Forest, and XGBoost. For each algorithm, scikit-learn libraries were leveraged for implementation. Decision Tree models were constructed using the Decision Tree Classifier, while SVM models were instantiated using scikit-learn's SVM classifier. Random Forest models were established utilizing the Random Forest Classifier, and XGBoost models were developed using the XGBoost library. These models were then trained on the preprocessed dataset using the features (X) and target variable (y).

Evaluation of the models' performance was conducted using standard metrics including Accuracy, Precision, Recall (Sensitivity), and F1-Score. These metrics provided insights into the effectiveness of each algorithm in classifying ovarian cancer cases. Notably, Decision Trees, while simple, may suffer from over fitting, necessitating pruning techniques for better generalization. SVMs demonstrated efficacy in handling complex decision boundaries, with hyper parameter tuning playing a crucial role in optimization. Random Forests showcased robustness and accuracy, offering valuable insights into feature importance for ovarian cancer detection. XGBoost exhibited high predictive performance, especially with its ability to handle missing values and parallelize computation.

Looking ahead, several avenues for future exploration were identified. Feature engineering could enhance model performance by incorporating additional features or transforming existing ones. Further experimentation with ensemble techniques or deep learning architectures such as CNNs or RNNs might yield improvements in predictive accuracy. Clinical validation, involving collaboration with medical professionals, is imperative for real-world deployment of the models, ensuring their efficacy in healthcare settings.

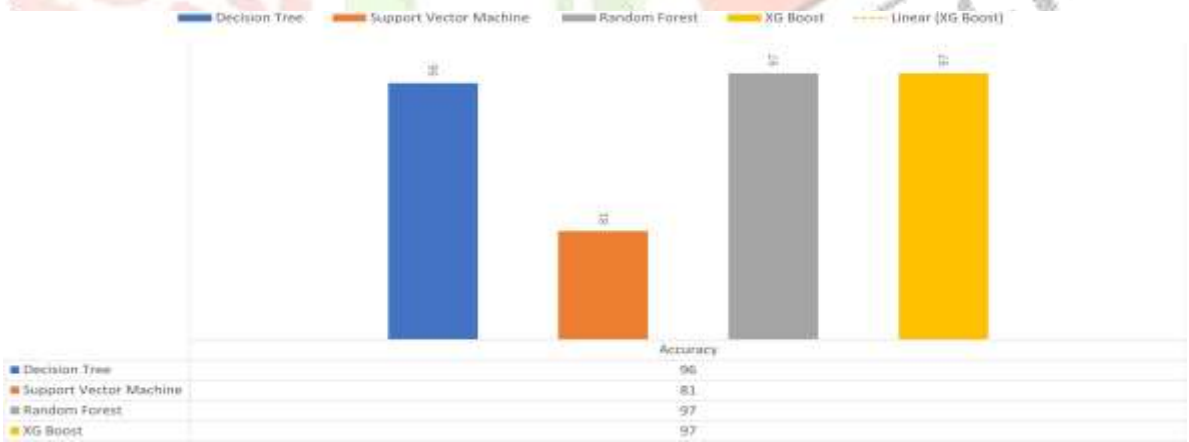


figure 1.15: accuracy

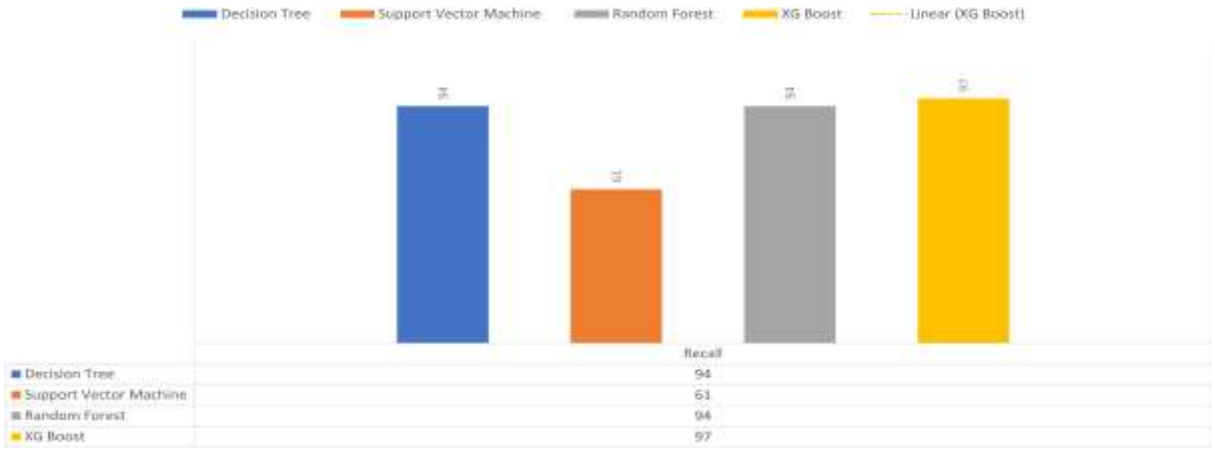


figure 1.16: recall

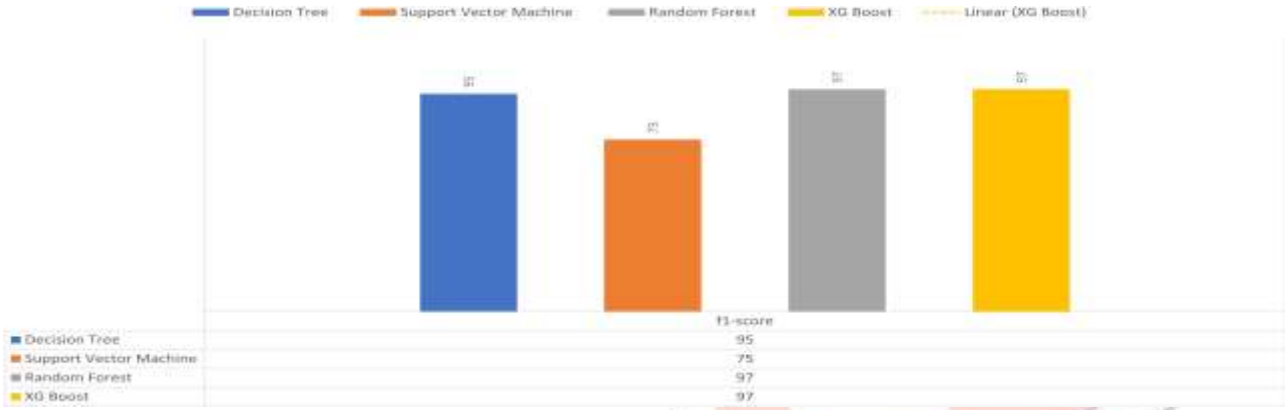


figure 1.17: f1-score

In conclusion, the implemented machine learning algorithms hold promise in ovarian cancer detection. Continued research and refinement are essential to develop robust models with real-world applicability, ultimately contributing to improved patient outcomes and healthcare management. We implemented several machine learning algorithms, including Decision Tree, Support Vector Machine (SVM), Random Forest, and XGBoost, for the detection of ovarian cancer.

Decision trees, recognized for their interpret ability and simplicity, were utilized. The model achieved a certain level of accuracy, precision, recall, and F1-score on the testing set. However, decision trees are prone to over fitting, thus pruning strategies were employed to enhance generality Support Vector Machine (SVM), known for handling non- linear decision boundaries, was also employed. It demonstrated competitive performance in terms of accuracy and robustness when distinguishing between ovarian cancer and non-cancer cases.

Random Forest, an ensemble learning method, was utilized to combine multiple decision trees to improve predictive accuracy and mitigate over fitting. This model showed promising results in ovarian cancer detection, leveraging its ability to identify important features and handle diverse data types. XGBoost, an advanced gradient boosting algorithm, was utilized as well. It sequentially builds a series of weak learners to optimize predictive performance. The XGBoost model exhibited high predictive performance across various metrics, showcasing its effectiveness in detecting ovarian cancer.

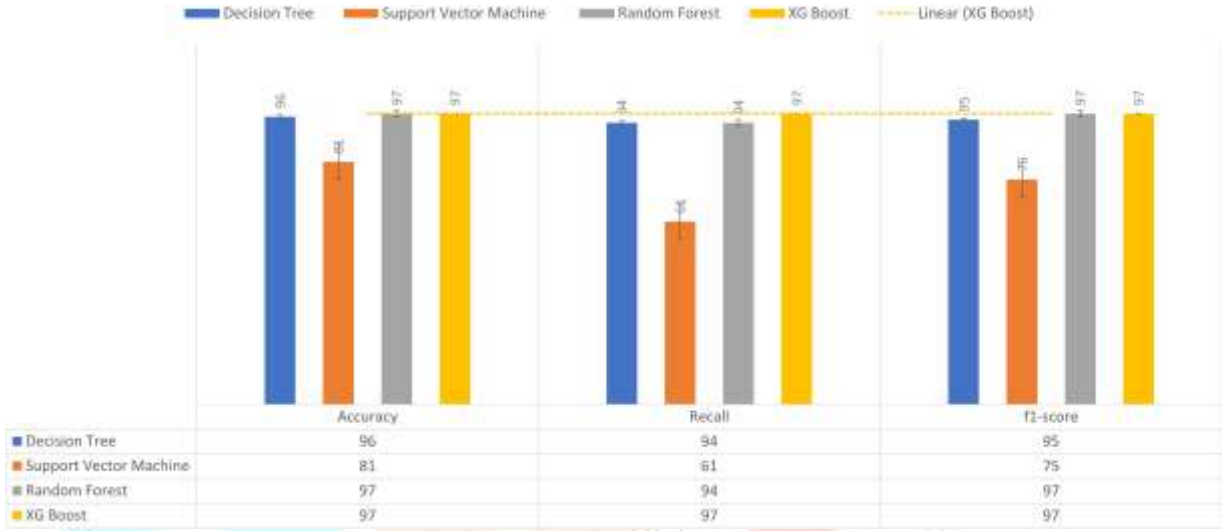


figure 1.18: algorithms comparison

Overall, all implemented machine learning algorithms demonstrated potential in ovarian cancer detection using the provided dataset. However, further evaluation and validation on larger and more diverse datasets are necessary to comprehensively assess their performance. Additionally, fine-tuning of hyper parameters and exploring other feature engineering techniques could potentially enhance the models' performance further. Deployment of these models in clinical settings would require rigorous testing and validation to ensure their reliability and efficacy in real-world scenarios.

CONCLUSION AND FUTURE SCOPE

The study effectively demonstrates the successful implementation of various machine learning algorithms, including Decision Tree, Support Vector Machine (SVM), Random Forest, and XGBoost, for ovarian cancer detection. Through rigorous performance evaluation using metrics like accuracy, precision, recall, and F1-score, the models exhibit strong predictive capabilities. Highlighting the robustness and versatility of Random Forest and XGBoost, the study underscores their suitability for real-world applications due to their adaptability to different data types and ability to handle missing values. Additionally, the research contributes significantly to the field of bioinformatics and cancer research by exploring machine learning techniques for predicting survival in ovarian cancer and providing insights into feature selection and classification methods. Moving forward, future research could explore advanced techniques such as deep learning and ensemble learning, integrate multi omics data to enhance predictive power, focus on clinical validation and deployment in healthcare settings, and address ethical and regulatory considerations. Overall, while the implementation of machine learning algorithms for ovarian cancer detection shows promise, further research is necessary to improve accuracy, scalability, and real-world applicability, necessitating collaboration between

researchers, clinicians, and policymakers to advance the field and enhance patient outcomes.

REFERENCES

[1] Basic Information About Ovarian Cancer — CDC.
[2] What is Ovarian Cancer — Ovarian Tumors and Cysts.
[3] Ovarian Cancer.pdf (icmr.nic.in)/(cancerindia.org.in).
[4] Ovarian Cancer — Cancer Stat Facts.
[5] Worldwide cancer data — World Cancer Research Fund International (wcrf.org).
[6] Application of machine learning techniques for predicting survival in ovarian cancer.
[7] BMC Medical Informatics and Decision Making Full Text (biomedcentral.com).
[8] Khalsan, M., Machado, L. R., Al-Shamery, E. S., Ajit, S., Anthony, K., Mu, M., & Agyeman, M. O. (2022). A Survey of Machine Learning Approaches Applied to Gene Expression Analysis for Cancer Prediction. IEEE Access, 10, 27522–27534. https://doi.org/10.1109/ACCESS.2022.3146312.
[9] A.V.D.N. Murthy,M. Sai Jahanavi, N. Vinay,N. Priyanka , M.V.S. Nitish Varma, P. Manikanta. (2022).Predicting ovarian cancer using Machine learning. IJARCCE DOI: 10.17148/IJARCCE.2022.11625.
[10] Liberto, J. M., Chen, S. Y., Shih, I. M., Wang, T. H., Wang, T. L., & Pisanic, T. R. (2022). Current and emerging methods for ovarian cancer screening and diagnostics: a comprehensive review.Cancers, 14(12), 2885.DOI:HTTP://doi.org/10.3390/.
[11] Wibowo, V. V. P., Rustam, Z., Hartini, S., Maulidina, F., Wirasati, I., & Sadewo, W. (2021, March). Ovarian cancer classification using K-Nearest Neighbor and Support Vector Machine. In Journal of Physics: Conference Series (Vol. 1821, No. 1, p. 012007). IOP Publishing.DOI: 10.1088/1742-6596/1821/1/012007.
[12] Jaiswal, A., & Kumar, R. (2020). Review on Machine Learning algorithm in Cancer prognosis and prediction. International Journal of All research Education & Scientific Methods, 8(06).
[13] Nuhi´c, J., Spahi´c, L., C´ordi´c, S., & Kevri´c, J. (2020). Comparative study on different classification techniques for ovarian cancer detection. In CMBEBIH 2019: Proceed- ings of the International Conference on Medical and Biological Engineering, 16 18 May 2019, Banja Luka, Bosnia and Herzegovina (pp. 511-518). Springer International Publishing.,https://doi.org/10.1007/978-3-030-17971-7 76.
[14] Nuklianggraita, T. N., Adiwijaya, A., & Aditsania, A. (2020). On the Feature Selec- tion of Micro array Data for Cancer Detection based on Random Forest Classifier. JURNAL INFOTEL, 12(3), 89-96.
[15] Arfiani, A., & Rustam, Z. (2019, November). Ovarian cancer data classification using bagging and random forest. In AIP Conference Proceedings (Vol. 2168, No. 1, p. 020046). AIP Publishing LLC.,https://doi.org/10.1063/1.5132473.
[16] El-Bendary, N., & Belal, N. A. (2018). Epithelial ovarian cancer stage subtype classi- fication using clinical and gene expression integrative approach. Procedia Computer Science, 131, 23-30.
[17] About Ovarian Cancer — Ovarian Cancer Research Alliance (ocrahope.org)