OVARIAN CANCER PREDICTION IN EARLY STAGE USING MACHINE LEARNING APPROACHES

Submitted for partial fulfillment of the requirements

for the award of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE & ENGINEERING

by

Panidapu Hima Chandana	-	**19BQ1A05H1**
Perumalla Hema Sri	-	**19BQ1A05H7**
Navuluri Nitish Kumar	-	**19BQ1A05G2**
Nallamothu Hemanth	-	**19BQ1A05F8**

Under the guidance of

Dr. K. Lohitha Lakshmi, MTech, Ph.D.

Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

(B. Tech Program is Accredited by NBA)

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Permanently Affiliated to JNTU Kakinada, Approved by AICTE

Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified

NAMBUR (V), PEDAKAKANI (M), GUNTUR – 522 508

Tel no: 0863-2118036, url: www.vvitguntur.com

April 2023

DECLARATION

We, Ms. Panidapu Hima Chandana, Ms. Perumalla Hema Sri, Mr. Navuluri Nitish Kumar, Mr. Nallamothu Hemanth, hereby declare that the Project Report entitled **"Ovarian Cancer Prediction in Early Stage Using Machine Learning Approaches"** done by us under the guidance of Dr. K. Sri Lohitha Lakshmi, Associate Professor, CSE at Vasireddy Venkatadri Institute of Technology is submitted for partial fulfillment of the requirements for the award of Bachelor of Technology in Computer Science & Engineering. The results embodied in this report have not been submitted to any other University for the award of any degree.

DATE :

PLACE :

SIGNATURE OF THE CANDIDATE (S)

P. Hima Chandana :

P. Hema Sri :

N. Nitish Kumar :

N. Hemanth :

ACKNOWLEDGEMENT

We take this opportunity to express my deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand my ideas and helped me towards the successful completion of this project.

First and foremost, we express our deep gratitude to **Mr. Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech programme.

We express our sincere thanks to **Dr. Y. Mallikarjuna Reddy**, Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech programme.

We express our sincere gratitude to **Dr. V. Ramachandran**, Professor & HOD, Computer Science & Engineering, Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand my ideas.

We would like to express our sincere gratitude to my guide **Dr. K. Lohitha Lakshmi**, Associate Professor, CSE for her insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to express our sincere heartfelt thanks to our Project Coordinator **Dr. N. Sri Hari**, Associate Professor, CSE for his valuable advices, motivating suggestions, moral support, help and coordination among us in successful completion of this project.

We would like to take this opportunity to express our thanks to the **teaching and non-teaching staff** in the Department of Computer Science & Engineering, VVIT for their invaluable help and support.

P. Hima Chandana - 19BQ1A05H1

P. Hema Sri - 19BQ1A05H7

N. Nitish Kumar - 19BQ1A05G2

N. Hemanth - 19BQ1A05F8

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY
Permanently Affiliated to JNTU Kakinada, Approved by AICTE
Accredited by NAAC with 'A' Grade, ISO 9001:2008 Certified
Nambur, Pedakakani (M), Guntur (Dt) - 522508
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
B.Tech Program is Accredited by NBA

CERTIFICATE

This is to certify that this **Project Report** is the bonafide work of **Ms. Panidapu Hima Chandana, Ms. Perumalla Hema Sri, Mr. Navuluri Nitish Kumar, Mr. Nallamothu Hemanth**, bearing Reg. No. **19BQ1A05H1, 19BQ1A05H7, 19BQ1A05G2, 19BQ1A05F8** respectively who had carried out the project entitled **"Ovarian Cancer Prediction In Early Stage Using Machine Learning Approaches"** under our supervision.

Project Guide
(Dr. K. Lohitha Lakshmi, Associate Professor)

Head of the Department
(Dr. V. Ramachandran, Professor)

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

INDEX

TABLE OF CONTENTS

LIST OF FIGURES

NOMENCLATURE

RF	Random Forest
LR	Logistic Regression
GBM	Gradient Boosting Machine
LGBM	Light Gradient Boosting Machine
XGBM	Extreme Gradient Boosting Machine
DNA	Deoxyribonucleic acid
BRCA1	BReast CAncer gene 1
BRCA2	BReast CAncer gene 2
SVM	Support Vector Machine
DT	Decision Tree
ROC	Receiver Operating Characteristic
ANN	Artificial Neural Network
CA 125	Carbohydrate Antigen 125
DCNN	Diffusion-Convolutional Neural Network
CT	Computed Tomography
MRI	Magnetic Resonance Imaging
MDP	Markov Decision Process
GOSS	Gradient-based One Side Sampling
EFB	Exclusive Feature Bundling
PIDD	Pima Indian Diabetes Database
TCGA	The Cancer Genome Atlas Research Network
SVMSMOTE	Support Vector Machine Synthetic Minority Oversampling TEchnique
PPV	Positive Predictive Value
LASSO	Least Absolute Shrinkage and Selection Operator
GBDT	Gradient Boosting Decision Tree
TVS	Trans Vaginal Sonography
ML	Machine Learning

ABSTRACT

Ovarian cancer is the irregular growth of cells that forms as a lump in the ovaries. These cells multiply quickly and can destroy healthy body tissue. Ovarian cancer is a global problem, is typically diagnosed at a late stage due to non-effective screening strategy. Still, there are no proper treatments that can adequately cure this disease. However, early-stage detection could reduce mortality rate of the patients. The main aim of our project is to apply machine learning models along with statistical methods over clinical data obtained from individual patients to conduct predictive analytics for early detection. In statistical analysis mutual information test plays an important role to find the significant biomarkers. A set of machine learning models including Random Forest (RF), Extreme Gradient Boosting Machine (XGBoost), Logistic Regression (LR), Gradient Boosting Machine (GBM) and Light Gradient Boosting Machine (LGBM) are used in classification to classify the benign and malignant ovarian cancer. By using proposed system, it can significantly identify the class of benign and malignant patients. The data collected is analyzed and pre-processed before it is used for model training and testing. Machine learning detection could play a good role in cancer diagnosis since early-stage detection is typically not available.

Keywords: Biomarkers, Mutual Information, Ovarian Cancer, Predictive Models, Risk Factors.

CHAPTER 1
INTRODUCTION

1.1 WHAT IS OVARIAN CANCER?

Cancer is a large group of diseases that can start in almost any organ or tissue of the body when abnormal cells grow uncontrollably, go beyond their usual boundaries to invade adjoining parts of the body and/or spread to other organs. The latter process is called metastasizing and is a major cause of death from cancer. A neoplasm and malignant tumour are other common names for cancer. Ovarian cancer is a cancerous tumour that forms in the tissues of an ovary. The ovaries are a pair of female reproductive glands that make eggs and female hormones.

Ovarian cancer is an abnormal mass of ovarian tissue whose growth outpaces and is not coordinated with the normal tissue. It also continues to grow excessively even after the initial stimulus that caused the alteration has stopped. It is crucial to predict ovarian cancer at an early stage because the mortality rate of patients is rising daily. Bloating or swelling in the abdomen, pain in the pelvis, difficulty eating or feeling full quickly, and frequent urination are all signs of ovarian cancer. Age, certain genetic mutations, and a family history of the disease are all risk factors for ovarian cancer.

There are three types of ovarian cancers: epithelial ovarian carcinomas, germ cell tumors, and stromal cell tumors.

Epithelial ovarian carcinomas: These are the most common type of ovarian cancer. About 85% to 90% of these cancers involve the cells that cover the outer surface of the ovary. They commonly spread first to the lining and organs of the pelvis and abdomen and then to other parts of the body. Nearly 70% of women with this type of ovarian cancer are diagnosed in the advanced stages.

Germ cell tumors: These make up less than 2% of all ovarian cancers. They begin in the reproductive cells that are a woman's "eggs". Ninety percent of patients with germ cell tumors survive five years after diagnosis. Teenagers and women in their 20s are more likely to develop this type of ovarian cancer.

Stromal cell tumors: These represent about 1% of all ovarian cancers. They form in the tissues that support the ovaries. This type of cancer is often found in the early stages.

The most common type is epithelial cancer. It begins in the cells that cover the ovary.There are also two related types of epithelial cancer that can spread to the ovaries:

- Fallopian tube cancer forms in the tissue lining a fallopian tube. The fallopian tubes are a pair of long, slender tubes on each side of the uterus. The uterus is the female reproductive organ where a baby grows during pregnancy.

- Primary peritoneal cancer forms in the tissue lining the peritoneum. Your peritoneum is a tissue lining that covers the organs in the abdomen (belly).

These two cancers are similar to ovarian cancer, and they have the same treatments. So some medical experts also consider those two types as ovarian cancer.

1.2 CAUSES OF OVARIAN CANCER?

Ovarian cancer happens when there are changes mutations in generic materials (DNA). Often, the exact cause of these genetic changes is unknown. Most ovarian cancers are caused by genetic changes that happen during your lifetime. But sometimes these genetic changes are inherited, meaning that you are born with them. Ovarian cancer that is caused by inherited genetic changes is called hereditary ovarian cancer.

There are also certain genetic changes that can raise your risk of ovarian cancer, including changes called BReast Cancer gene 1(BRCA1) and BReast CAncer gene 2(BRCA2). These two changes also raise your risk of breast and other cancers. Besides genetics, lifestyle and the environment can affect your risk of ovarian cancer.

Some of the causes are:

- Have a family history of ovarian cancer in a mother, daughter, or sister
- Have inherited changes in the BRCA1 or BRCA2 genes.
- Have certain other genetic conditions, such as Lynch syndrome
- Took hormone replacement therapy
- Are overweight or have obesity
- Are tall
- Are older, especially those who have gone through menopause

1.3 SYMPTOMS OF OVARIAN CANCER:

Ovarian cancer may not cause early signs or symptoms. By the time you do have signs or symptoms, the cancer is often advanced.

The signs and symptoms may include:

- Pain, swelling, or a feeling of pressure in the abdomen or pelvis
- Sudden or frequent urge to urinate
- Trouble eating or feeling full
- A lump in the pelvic area
- Gastrointestinal problems, such as gas, bloating, or constipation

1.4 RISK FACTORS

Factors that can increase your risk of ovarian cancer include:

- **Inherited gene changes.** A small percentage of ovarian cancers are caused by genes changes you inherit from your parents. The genes that increase the risk of ovarian cancer include BRCA1 and BRCA2. These genes also increase the risk of breast cancer.

- **Family history of ovarian cancer.** If you have blood relatives who have been diagnosed with ovarian cancer, you may have an increased risk of the disease.

- **Being overweight or obese.** Being overweight or obese increases the risk of ovarian cancer.

- **Postmenopausal hormone replacement therapy.** Taking hormone replacement therapy to control menopause signs and symptoms may increase the risk of ovarian cancer.

- **Endometriosis.** Endometriosis is an often painful disorder in which tissue similar to the tissue that lines the inside of your uterus grows outside your uterus.

- **Age when menstruation started and ended.** Beginning menstruation at an early age or starting menopause at a later age, or both, may increase the risk of ovarian cancer.

CHAPTER 2

LITERATURE SURVEY

Md.Martuza Ahamad et al. [1] Early stage detection of ovarian cancer can increase the life span of patients. The data set consists of 349 patients clinical data and 49 features. Over them 171 were the ovarian cancer patients and 178 were the ovarian tumor patients. The whole data set was divided into 3 subgroups namely general chemistry, blood routine test and tumor marker. The set of machine learning models used are Random Forest (RF), Support Vector Machine (SVM), Decision Tree (DT), Extreme Gradient Boosting Machine (XGBoost), Logistic Regression (LR), Gradient Boosting Machine (GBM) and Light Gradient Boosting Machine (LGBM). Those are used to develop and build the classification models that divides benign and malignant ovarian cancer patients. The most significant biomarkers observed are carbohydrate antigen 125, carbohydrate antigen 19-9, carcino-embryonic antigen, and human epididymis protein 4. The results obtained from Random Forest(RF), Gradient Boosting Machine(GBM), Light Gradient Boosting Machine (LGBM) classifiers showed high level of accuracy.

Arcuda et al. employed serum proteome profile data to drive wavelet feature selection in machine learning methods.

Patrick et al. made efforts to predict ovarian cancer using regularized logistic regression[2]. 349 medical records from the “Third Affiliated Hospital of Soochow University” make up the data set used in this study. A logistic regression model is constructed with a Least Absolute Shrinkage and Selection Operator (LASSO) regularization penalty. Accuracy of 90.6% is obtained using Logistic regression.

SuthamerthiElavarasu et al. made a review on Machine Learning Applications in Ovarian Cancer Prediction[3]. Three principle approaches are stated for the selection of features, namely integrated, filters and envelopes approach. The survey concludes that by the analysis of various studies for predicting the outcomes of disease, the machine learning techniques and classification algorithms provides useful tools.

Viji Vinod et al. considered the TCGA ovarian cancer database of gene expression values. Machine learning model SVM showed the highest accuracy for recurrence and survival predictions. Yang et al. showed that the decision tree combined with Support Vector Machine-Synthetic Minority Oversampling TEchnique (SVMSMOTE) showed the top PPV (Positive Predictive Value) with 0.9041[4].

Munetoshi et al. had established that the histo-pathological identification of ovarian cancer may be predicted using artificial intelligence from preoperative assessment [5]. Lu et al. decided to evaluate 49 features from patient characteristics to build machine learning models. For performing modeling procedure they had used the combination of method named Minimum Redundancy Maximum Relevance (MRMR) feature selection, ReliefF feature selection also decision tree analysis. Finally they found that a prediction accuracy of 92.1% through decision tree approach using HE4 and carcino-embryonic antigen (CEA)[7].

Many studies have used CA125 as a marker of ovarian tumor[8]. To differentiate benign and malignant tumor In 1990, Jacobs et al. used the following factors- age, ultrasound status, clinical feature, menstrual history ,CA-125 levels as features to distinguish between benign and malignant ovarian tumors. Their experiment yielded a sensitivity of 81% and specificity of 75%.

He has used the data set consisting of 202 patients information from preoperative examinations. Highest accuracy of 80% was obtained using XGboost machine learning model.

R. Kasture et al. had used the histopathological images for the prediction of ovarian cancer[9]. Deep Learning method DCNN is used for training and evaluation[10]. Results of which highest accuracy of 91% was obtained from the KK-net model.

CHAPTER 3
PROBLEM IDENTIFICATION

Ovarian cancer is the cause of death for patients with gynecologic cancers in the United States, and it is responsible for 5% of cancer-related deaths in women overall. More than 70% of patients with ovarian cancer are diagnosed with late-stage disease. Ovarian cancer occurs when abnormal cells in ovaries or fallopian tubes grow and multiply out of control. Ovarian Cancer symptoms include abdominal pain, changes in eating habits, increase in the size of abdomen. Common treatment mechanisms are surgery, chemotherapy, hormone therapy etc. Machine learning algorithms with new approaches have great potentialities in predicting disease progression and malignancy diagnosis.

Ovarian cancer is challenging to diagnose at an early stage because of non-specificity of the signs and symptoms. Since the symptoms are often vague and may be associated with other conditions. The problem with ovarian cancer is that it is often not recognized until it is advanced, making it more difficult to treat and less likely to have a successful outcome. As the incidence of ovarian cancer is increasing day by day, early detection can greatly improve the survival of women with ovarian cancer.

The significance of late-stage ovarian cancer diagnosis highlights the importance of early detection and the need for improved screening and diagnostic tools to improve patient outcomes. Often there are challenges for developing treatment for all ovarian cancer patients such as heterogeneity of ovarian cancer and also side effects of the treatment.

The usage of machine learning models is a broadly recognized mechanism for showing disease-related factors as distinguishing markers in predictive patient diagnostics. Here, the data set considered is preprocessed and going to implement machine learning algorithms to identify important features in early diagnosis of ovarian cancer patients.

CHAPTER 4

AIM & SCOPE

4.1 EXISTING SYSTEM

Here is the overview of existing methodology for ovarian cancer treatment:

1. Diagnosis: The first step in treating ovarian cancer is to diagnose the disease. This typically involves a physical examination, imaging tests (such as CT scans or MRI), blood tests, and biopsy (a procedure to remove a small sample of tissue from the ovary to be examined under a microscope).

2. Surgery: Surgery is usually the first line of treatment for ovarian cancer. The goal of surgery is to remove as much of the cancer as possible. The extent of surgery may depend on the stage and extent of the cancer. In some cases, the surgeon may remove one or both ovaries, the fallopian tubes, the uterus, and other nearby tissues or organs. In some cases, a laparoscopic approach may be used.

3. Chemotherapy: Chemotherapy is typically given after surgery to kill any remaining cancer cells. Chemotherapy drugs can be given intravenously or directly into the abdominal cavity (intraperitoneal chemotherapy). Chemotherapy is often given in cycles, with periods of treatment followed by periods of rest to allow the body to recover.

4. Radiation therapy: Radiation therapy may be used in some cases to kill cancer cells or to relieve symptoms such as pain. Radiation therapy is typically used in combination with surgery or chemotherapy.

5. Targeted therapy: Targeted therapy is a type of cancer treatment that targets specific proteins or genes that are involved in the growth and spread of cancer cells. Targeted therapy drugs can be given orally or intravenously.

6. Immunotherapy: Immunotherapy is a type of cancer treatment that works by stimulating the immune system to recognize and attack cancer cells. Immunotherapy drugs can be given orally or intravenously.

7. Clinical trials: Patients with ovarian cancer may be eligible to participate in clinical trials that are testing new treatments or treatment combinations.

Overall, the existing methodology for ovarian cancer treatment involves a combination of surgery, chemotherapy, radiation therapy, targeted therapy, and immunotherapy, tailored to the individual needs of the patient. It's important for patients to work closely with their medical team to determine the best treatment plan for their specific situation.

Screening strategies that are available are Trans-vaginal Sonography (TVS) and CA125, but neither is specific enough to identify cancer when used alone. As a result we proposed a solution for early prediction of ovarian cancer. For early prediction identification of bio markers plays a significant role. To identify the significant bio markers from the data set that helps in prediction we implemented a feature selection technique mutual information.

4.2 DOMAIN

Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allows computers to learn from past examples and to detect patterns from large or complex data sets. This capability is particularly well-suited to medical applications, especially those that depend on complex measurements. As a result, machine learning is frequently used in cancer diagnosis and detection. More recently machine learning has been applied to cancer prognosis and prediction. These techniques have therefore been used as a model for the development and treatment of cancer. Many of these methods are widely used for the development of predictive models for predicating a cure for cancer, some of the methods are artificial neural networks (ANNs), support vector machine (SVMs) and decision trees (DTs).

4.3 SCOPE

For the input data set, the machine learning models which were going to applied are Random Forest (RF), Extreme Gradient Boosting Machine (XGBoost), Logistic Regression (LR), Gradient Boosting Machine (GBM) and Light Gradient Boosting Machine (LGBM). Random Forest (RF) considers training data arbitrarily to handle the over-fitting problems in an efficient way. Gradient Boosting Machine (GBM) is an ensemble learning method that merges multiple learners to make a robust one through the optimization. Logistic Regression (LR) is implemented in statistics used in binary classification problems. Light Gradient Boosting Machine (LGBM) is an improved version of GBM depending on tree-based learning techniques, handle a massive volume of data and perform at a high-accuracy level with limited computing resources. The prediction model that was going to develop divides the data set into sub groups, uses Accuracy, Precision, Recall, F1-score and log-loss evaluation metrics for testing the classifier performance and to obtain highest accuracy classifier the results are being compared.

CHAPTER 5
MACHINE LEARNING

5.1 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", in order to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks. Machine learning plays a critical role in identifying diseases by analyzing large amounts of data to identify patterns and make predictions. The below are some of the key reasons why machine learning is important in disease identification:

- Early detection: Machine learning algorithms can be trained to analyze medical images or other diagnostic tests to detect diseases at an early stage when they are most treatable.
- Improved accuracy: Machine learning algorithms can analyze large amounts of data more accurately and consistently than humans, reducing the risk of misdiagnosis or missed diagnoses.
- Personalized medicine: Machine learning algorithms can analyze an individual's genetic and medical history to develop personalized treatment plans that are tailored to their unique needs.
- Efficient screening: Machine learning algorithms can be used to screen large populations for diseases, identifying individuals who are at high risk and referring them for further testing or treatment.
- Rapid diagnosis: Machine learning algorithms can analyze data in real-time, enabling rapid diagnosis and treatment of diseases.
- Cost savings: By identifying diseases at an early stage or preventing their onset altogether, machine learning can potentially save healthcare systems significant costs associated with treating advanced-stage diseases.

Overall, machine learning is important in disease identification because it enables early detection, improves accuracy, enables personalized medicine, makes screening more efficient, enables rapid diagnosis, and can potentially save healthcare systems significant costs.

Also Machine learning had a significant influence in the medical field in recent years. The below are some of the ways machine learning has impacted medicine:

- Medical imaging: Machine learning algorithms have been developed that can analyze medical images such as X-rays, CT scans, and MRI scans. These algorithms can assist in the early detection of diseases and the identification of abnormalities that may be missed by human analysis.
- Predictive analytics: Machine learning can be used to analyze large amounts of medical data to identify patterns and predict outcomes. This can be used to identify patients who are at risk of developing certain diseases or to predict the likelihood of a patient responding to a particular treatment.
- Electronic health records: Machine learning algorithms can be used to analyze electronic health records to identify trends and patterns that can inform clinical decision-making.
- Drug discovery: Machine learning algorithms can be used to analyze large datasets to identify potential drug targets and to predict the efficacy of new drugs.
- Personalized medicine: Machine learning can be used to analyze an individual's genetic and medical history to develop personalized treatment plans.
- Remote monitoring: Machine learning can be used to monitor patients remotely, analyzing data from wearable devices or sensors to detect changes in health status and provide early intervention when necessary.

Overall, machine learning has the potential to revolutionize the medical field, making diagnosis, treatment, and monitoring more accurate, efficient, and personalized. However, it's important to note that machine learning algorithms are not perfect and must be carefully developed and validated to ensure their accuracy and safety.

5.2 MACHINE LEARNING APPROACHES:

Early classifications for machine learning approaches sometimes divided them into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system.

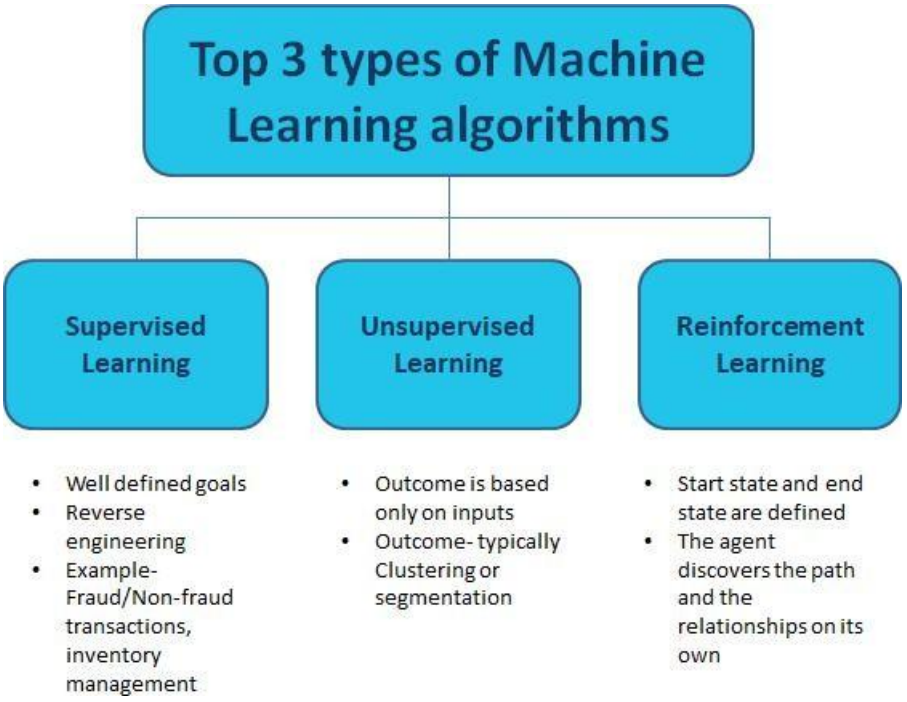

Fig 5.1 Types of Machine Learning Algorithms

5.2.1 ***Supervised learning:*** Supervised learning algorithms build a mathematical model of a set of data that contains both the inputs and the desired outputs. The data is known as training data, and consists of a set of training examples. Each training example has one or more inputs and the desired output, also outputs known as a supervisory signal. In the mathematical model, each training example is represented by an array or vector, sometimes called a feature vector, and the training data is represented by a matrix.

5.2.2 ***Unsupervised learning:*** Unsupervised learning algorithms take a set of datathat contains only inputs, and find structure in the data, like grouping or clustering of data points. The algorithms, therefore, learn from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning algorithms identify commonalities in the data and react based on the presence or absence of such commonalities in each new piece of data. A central application of unsupervised learning is in the field of density estimation in statistics, such as finding the probability density function. Though unsupervised learning encompasses other domains involving summarizing and explaining data features.

5.2.3 ***Semi-Supervised Learning:*** Semi-supervised learning falls between unsupervised learning (without any labeled training data) and supervised learning (with completely labeled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabeled data, when used in conjunction with a small amount of labeled data, can produce a considerable improvement in learning accuracy. In weakly supervised learning, the training labels are noisy, limited, or imprecise; however,these labels are often cheaper to obtain, resulting in larger effective training sets.

5.2.4 ***Reinforcement learning:*** Reinforcement learning is an area of machine learning concerned with how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward. Due to its generality, the field is studied in many other disciplines, such as game theory, control theory, operations research, information theory, simulationbased optimization, multi-agent systems, swarm intelligence, statistics and genetic algorithms. In machine learning, the environment is typically represented as a Markov Decision Process (MDP). Many reinforcement learning algorithms use dynamic programming techniques. Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible.

5.2.5 Other Learning:

- Feature Learning
- Sparse Dictionary Learning
- Artificial Neural Networks
- Decision Trees
- Support Vector Machines
- Regression Analysis
- Genetic Algorithms
- Training models
- Federated Learning

5.3 MACHINE LEARNING MODELS

The Machine Learning Models implemented were Gradient Boosting, Extreme Gradient Boosting , Light Gradient Boosting , Logistic Regression and Random Forest.

5.3.1 GRADIENT BOOSTING

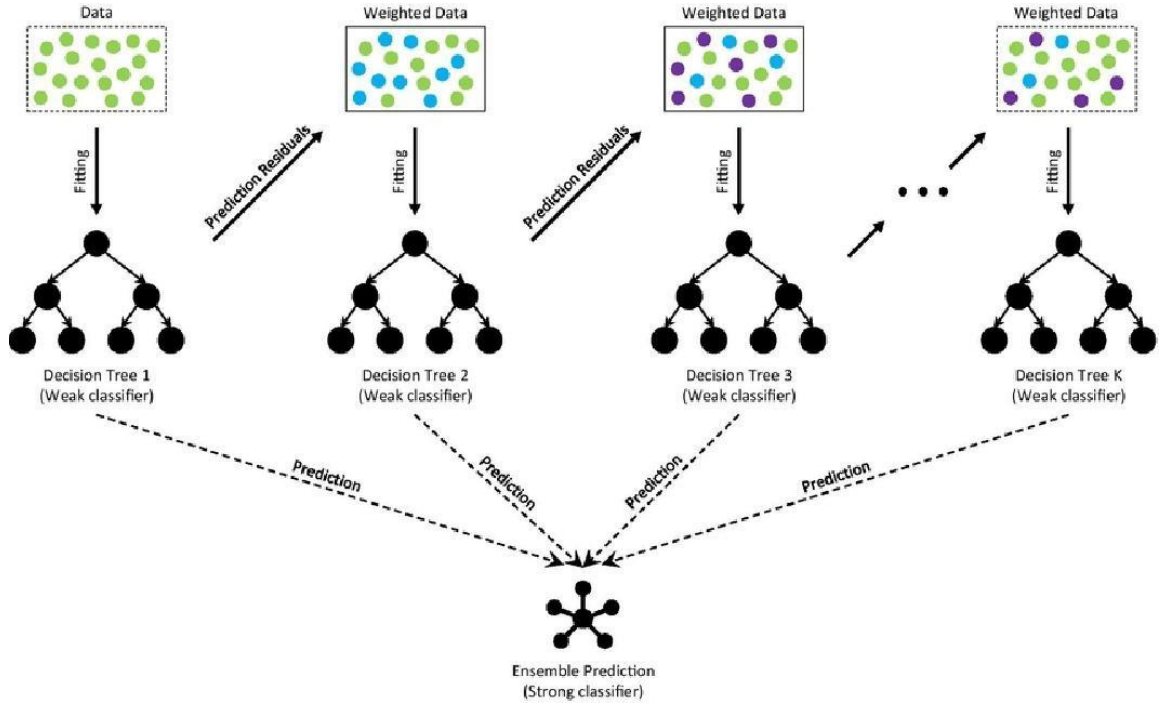

Figure 5.2 The Architecture Of Gradient Boosting Decision Tree

Basic Steps for performing gradient boosting:

i. Fit a decision tree on data. [call x as input and y as output]

ii. Calculate error residuals. Actual target value, minus predicted target value [e1= y – y_predicted1]

iii. Fit a new model on error residuals as target variable with same input variables [call it e1_predicted]

iv. Add the predicted residuals to the previous predictions [y_predicted2 = y_predicted1 + e1_predicted]

v. Fit another model on residuals that is still left. i.e. [e2 = y – y_predicted2] and repeat steps 2 to 5 until it starts over fitting or the sum of residuals become constant. Overfitting can be controlled by consistently checking accuracy on validation data.

Gradient Boosting Machine is a machine learning technique which was used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees.When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function.

- Gradient boosting is a machine learning technique that makes the prediction work simpler.
- It can be used for solving many daily life problems. However, boosting works best in a given set of constraints & in a given set of situations.
- The three main elements of this boosting method are a loss function, a weak learner, and an additive model.
- The regularization technique is used to reduce the over-fitting effect.
- An aspect of gradient boosting is regularization through shrinkage. If the learning rates are less than 0.1, it is very important to generalize the prediction model.
- Gradient boosting creates prediction-based models in the form of a combination of weak prediction models.
- Weak hypotheses are parameters whose performance is slightly better than the randomly made choices.
- Leo Breiman, an American Statistician, interpreted that boosting can be an optimization algorithm when used with suitable cost functions.
- One does optimization of cost functions by iteratively picking up the weak hypotheses or a function with a relatively negative gradient.
- The gradient boosting method has witnessed many further developments to optimize the cost functions.
- The working of gradient boosting revolves around the three main elements. They are a loss function, a weak learner, an additive model.

5.3.2 EXTREME GRADIENT BOOSTING MODEL

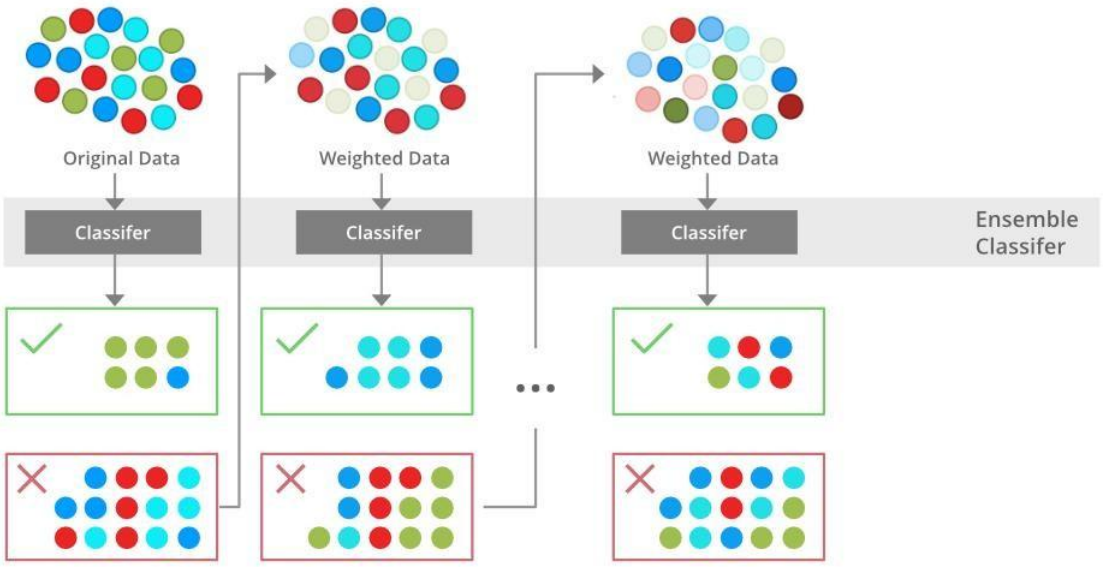

Figure 5.3 Extreme Gradient Boosting Machine Model

a. **XGBoost** is an implementation of Gradient Boosted decision trees. In this algorithm, decision trees are created in sequential form.

b. Weights play an important role in XGBoost. Weights are assigned to all the independent variables which are then fed into the decision tree which predicts results.

c. The weight of variables predicted wrong by the tree is increased and the variables are then fed to the second decision tree.

d. These individual classifiers/predictors then ensemble to give a strong and more precise model. It can work on regression, classification, ranking, and user-defined prediction problems.

Gradient Boosting Machines (GBMs) and Extreme Gradient Boosting Machines (XGBMs) are both powerful machine learning models used for building predictive models. They are both ensemble models that combine multiple weak models to create a strong predictive model. However, there are some key differences between the two models:

Regular Gradient Boosting Machines (GBMs) use gradient descent optimization to minimize a loss function while training. Extreme Gradient Boosting Machines (XGBMs) add a "regularization" term to the loss function to prevent overfitting, which helps improve the model's generalization ability.

XGBMs use a more advanced tree-building algorithm than GBMs. Specifically, XGBMs use a technique called "approximate greedy algorithm" which makes the tree-building process more efficient and reduces the training time.

XGBMs have additional hyperparameters that can be tuned to further improve the model's performance, such as "gamma" which controls the minimum loss reduction required to make a further partition on a leaf node, "lambda" which controls the L2 regularization term on weights, and "alpha" which controls the L1 regularization term on weights. In other words XGBMs are an improvement over GBMs and are known for their high accuracy and speed in training and prediction.

5.3.3 LIGHT GRADIENT BOOSTING MODEL

LightGBM is a gradient boosting framework based on decision trees to increases the efficiency of the model and reduces memory usage.
It uses two novel techniques: **Gradient-based One Side Sampling** and **Exclusive Feature Bundling (EFB)** which fulfills the limitations of histogram-based algorithm that is primarily used in all GBDT (Gradient Boosting Decision Tree) frameworks. The two techniques of GOSS and EFB described below form the characteristics of LightGBM Algorithm. They comprise together to make the model work efficiently and provide it a cutting edge over other GBDT frameworks.

Gradient-based One Side Sampling Technique for LightGBM:
Different data instances have varied roles in the computation of information gain. The instances with larger gradients(i.e., under-trained instances) will contribute more to the information gain. GOSS keeps those instances with large gradients (e.g., larger than a predefined threshold, or among the top percentiles), and only randomly drop those instances with small gradients to retain the accuracy of information gain estimation. This treatment can lead to a more accurate gain estimation than uniformly random sampling, with the same target sampling rate, especially when the value of information gain has a large range.

Exclusive Feature Bundling Technique for LightGBM:
High-dimensional data are usually very sparse which provides us a possibility of designing a nearly lossless approach to reduce the number of features. Specifically, in a sparse feature space, many features are mutually exclusive, i.e., they never take nonzero values simultaneously. The exclusive features can be safely bundled into a single feature (called an Exclusive Feature Bundle)

LightGBM splits the tree leaf-wise as opposed to other boosting algorithms that grow tree level-wise. It chooses the leaf with maximum delta loss to grow. Since the leaf is fixed, the leaf-wise algorithm has lower loss compared to the level-wise algorithm. Leaf-wise tree growth might increase the complexity of the model and may lead to overfitting in small datasets.

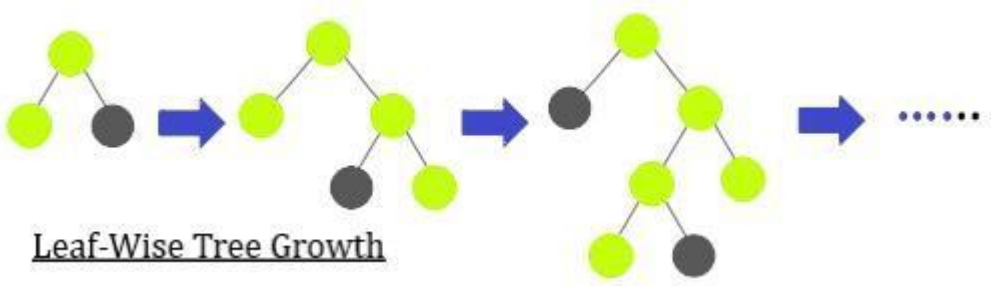

Figure 5.4 Light Gradient Boosting Model

a) LightGBM creates decision trees that grow leaf wise, which means that given a condition, only a single leaf is split, depending on the gain.

b) Leaf-wise trees can sometimes over fit especially with smaller datasets. Limiting the tree depth can help to avoid over fitting.

c) LightGBM uses a histogram-based method in which data is bucketed into bins using a histogram of the distribution.

d) The bins, instead of each data point, are used to iterate, calculate the gain, and split the data. This method can be optimized for a sparse data set as well.

e) Another characteristic of LightGBM is exclusive feature bundling in which the algorithm combines exclusive features to reduce dimensionality, making it faster and more efficient.

f) Gradient-based One Side Sampling (GOSS) is used for sampling the dataset in LightGBM.

g) GOSS weights data points with larger gradients higher while calculating the gain. In this method, instances that have not been used well for training contribute more.

h) Data points with smaller gradients are randomly removed and some are retained to maintain accuracy.

5.3.4 LOGISTIC REGRESSION

- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.

- In Logistic regression, instead of fitting a regression line, an "S" shaped logistic function is used, which predicts two maximum values (0 or 1).

- To implement the Logistic Regression the below steps are to be followed:

    - Data Pre-processing step
    - Fitting Logistic Regression to the Training set
    - Predicting the test result
    - Test accuracy of the result(Creation of Confusion matrix)
    - Visualizing the test set result.

Logistic Function

Logistic regression is named for the function used at the core of the method, the logistic function.

The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

1 / (1 + e^-value)

Where e is the base of the natural logarithms (Euler's number or the EXP() function) and value is the actual numerical value that we want to transform.

5.3.5 RANDOM FOREST

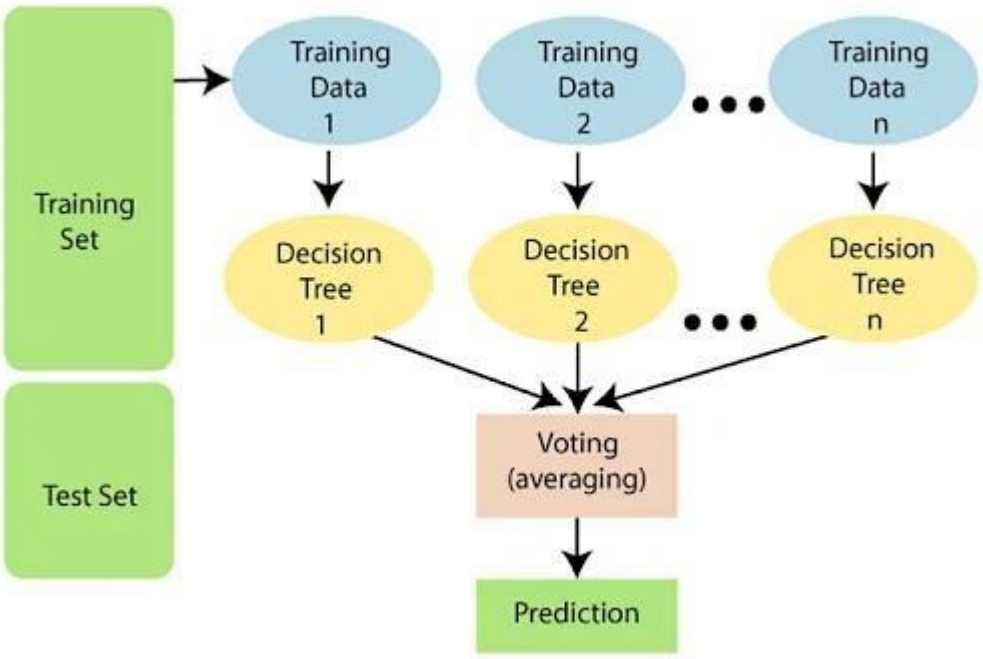

Figure 5.5 Illustration of Random Forest Model

- Random Forest is a classifier that contains a number of decision trees on various subsets of the given data set and takes the average to improve the predictive accuracy of that data set.
- Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of over-fitting.
- The following steps are the working of Random Forest Algorithm:

  Step 1: Select random samples from a given data or training set.

  Step 2: This algorithm will construct a decision tree for every training data.

  Step 3: Voting will take place by averaging the decision tree.

  Step 4: Finally, select the most voted prediction result is the final prediction result.

Random forest is one of the most popular algorithms based on the concept of ensemblelearning. It improves the result of complex problems by combining multiple learning models. The algorithm builds multiple decision trees and combines them to produce more accurate and stable results. The more the number of trees in the forest, the more accurate is the result.

Random forest algorithm builds a forest in the form of an ensemble of decision trees which adds more randomness while growing the trees. While splitting a node, the algorithm searches for the best features from the random subset of features which adds more diversity, thereby resulting in a better model.Therefore, for splitting a node, only a random subset of the features is taken into consideration. Instead of searching for the best possible threshold, we can also use random thresholds for each feature to build more random trees.

One of the best features of random forest is its simplicity in measuring the relative importance of each feature in the prediction. The more number of features one has, the more likely the model will suffer from over-fitting. One can decide which feature is not contributing to the prediction process and therefore, should be dropped by merely looking at the feature's importance.

Another great feature of the algorithm is versatility. Random forest can be used for both regression and classification tasks. Besides, viewing the relative importance that the algorithm assigns to the input features is also very easy. Convenience is another feature of the Random Forest algorithm since it often produces a great prediction result by using the default hyper-parameters. Besides, it is also very simple to understand the hyper parameters; there are not that many of them.

To get a more accurate prediction, one requires more trees. However, more trees slow down the model. This is one of the drawbacks of the random forest algorithm. Although they can be trained fast, these algorithms are quite slow in creating predictions. Due to a large number of trees, it becomes slow and ineffective in predicting real-time results. There are times and situations where run-time performance is more important, and therefore other approaches are preferred over the Random Forest Algorithm. Besides, random forest is a predictive modelling tool and not a descriptive tool. So, in cases where a description of the relationships of the data is required, other approaches are preferred over the random forest algorithm.

5.3.6 ENSEMBLE MACHINE LEARNING

A single algorithm may not make the perfect prediction for a given data set. Machine learning algorithms have their limitations and producing a model with high accuracy is challenging. Ensemble models are a machine learning approach to combine multiple other models in the prediction process. These models are referred to as base estimators. When we want to increase the performance of machine learning models ensembling learning techniques are used.For example to increase the accuracy of classification models or to reduce the mean absolute error ensembling is used. Ensembling also results in a more stable model.

Figure showing ensembling of models:

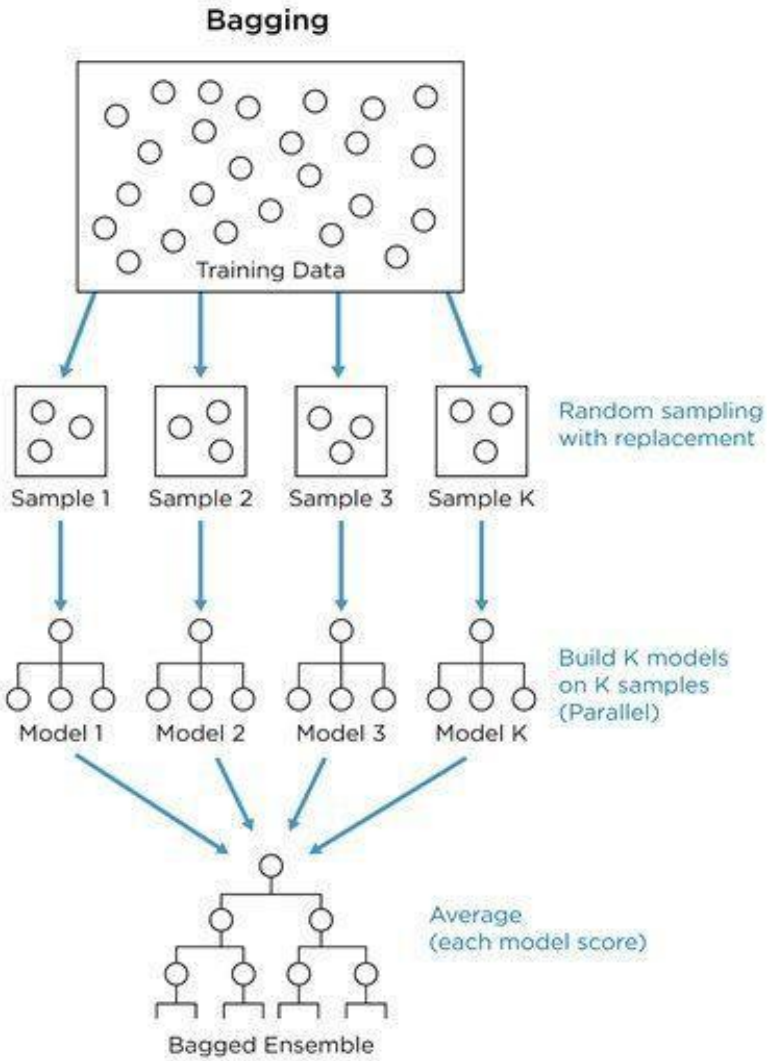

Figure 5.6 Ensembling Of Models

5.3.7 VOTING CLASSIFIER

Ensembling is a powerful technique to improve the performance of the model by combining various base models in order to produce an optimal and robust model. Types of Ensembling techniques include:

- Bagging or Bootstrap Aggregation
- Boosting
- Stacking Classifier
- Voting Classifier

A Voting Classifier in machine learning is an ensemble learning method that combines multiple base models (classifiers) to make a prediction. It works by aggregating the predictions of each base model and selecting the class that receives the majority vote as the final prediction.

There are two types of voting classifiers:

Hard Voting Classifier: In this type of classifier, the class with the highest number of votes is selected as the final prediction. Hard voting is also known as majority voting. The base model's classifiers are fed with the training data individually. The models predict the output class independent of each other. The output class is a class expected by the majority of the models.

Soft Voting Classifier: In this type of classifier, the class probabilities predicted by the base models are averaged and the class with the highest average probability is selected as the final prediction. In Soft voting, Classifiers or base models are fed with training data to predict the classes out of m possible courses. Each base model classifier independently assigns the probability of occurrence of each type. In the end, the average of the possibilities of each class is calculated, and the final output is the class having the highest probability

The importance of a Voting Classifier lies in its ability to improve the accuracy and stability of the model by combining the predictions of multiple base models. It can handle diverse and complex datasets by taking into account the strengths and weaknesses of each base model. Additionally, it can reduce over fitting and improve generalization by combining models with different biases. Also, a Voting Classifier can lead to better performance and more robust predictions, making it a valuable tool in machine learning.

CHAPTER 6

WORKING PROCESS

6.1 DATA SET

This data set is originally from the Third Affiliated Hospital of Suchow University. The objective is to predict based on clinical data whether a patient has ovarian cancer or not. Several constraints were placed on the selection of these instances from a larger database. Data set description is defined by Figure-6.1 and Figure-6.2 represents Attribute Description.

Figure-6.1: Data set description

Data set	No.of.Attributes	No.of.Instances
Third Affiliated Hospital of University of Soochow (public data set available in Kaggle)	49	349

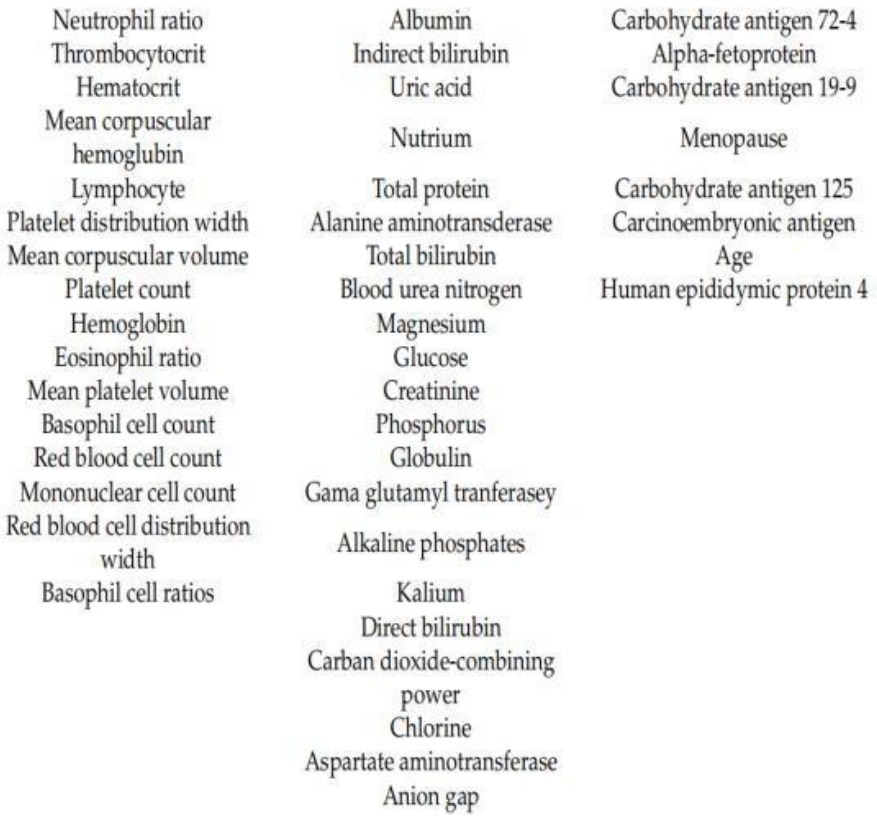

Figure 6.2 Attribute Description

6.2 DATA SCALING

We have applied data scaling, often referred to as feature scaling, to transform the values of the many attributes in our data set on a single scale. Many of automated learning techniques are sensitive to the scale of input data and may not operate as intended if the data is not scaled properly. Data scaling is done by standardizing the data by subtracting the mean from actual value and dividing with the standard deviation.

$$\text{Scaled Value} = (\text{actual value} - \text{mean}) / \text{deviation}$$

As a part of statistical analysis we have considered mutual information for feature selection.

6.3 FEATURE SELECTION

The process of selecting the pertinent characteristics from a huge data collection that contribute most to the prediction is known as feature selection. The fundamental objective of feature selection is to lessen model complexity and training time.

6.4 MUTUAL INFORMATION

A statistical entity known as mutual information measures the degree of dependency between two variables. In this context, mutual information is used to identify which features or biomarkers are more strongly associated with the presence of ovarian cancer. Mutual information between two variables X and Y can be calculated as

$$MI(X,Y) = H(X)+H(Y)-H(X|Y)$$

Where H(X) is the entropy of X, H(Y) is the entropy of Y and H(X|Y) is the joint entropy.

$$H(X) = -\sum p(X) \log p(X)$$

$$H(Y) = -\sum p(Y) \log p(Y)$$

6.5 WORKING OF THE PROPOSED SYSTEM

In the proposed system, first data set is collected from the Third Affiliated Hospital University of Soochow which is publicly available in the Kaggle and the data is pre-processed for scaling. The main aim of the proposed system is to identify the relevant features that are closely related to the target variable for which mutual information test is implemented along with we compare the models based on the evaluation metrics which are accuracy, recall, precision, f1-score based on these we identify the model which gives better performance in classifying the patients of benign and malignant tumour. In order to do so we have considered the models Logistic Regression, Random Forests, Gradient Boosting Machine, Extreme Gradient Boosting, Light Gradient Boosting and Voting Classifier. The results shown in the form of table comparing the evaluation metrics obtained for each model.

Step-1: Loading Data set

we have to import the data set by using **pandas library** for easily loading and manipulating data set.

To install pandas, use the following pip command: pip install pandas

Then we have to import it by using import pandas as pd, where “as pd” is alias name for pandas which we choosen.

In which it has the read_csv() function which is used to load the dataset i.e.,

```
ds = pd.read_csv('diabetes.csv')
```

where ds is variable which stores the dataset as data frames.

Step-2: Pre-processing:

In the pre-processing step we check for the missing values and the incorrect values to clean them because they lead to decrease in the accuracy of the algorithms.

The package **sklearn.preprocessing**provides several common utility functions and transformer classes to change raw feature vectors into a representation that is more suitable for the downstream estimators.

Standardization of datasets is a common requirement for many machine learning estimators implemented in scikit-learn; they might behave badly if the individual features do not more or less look like standard normally distributed data: Gaussian with zero mean and unit variance**.**

The **StandardScaler** assumes your data is normally distributed within each feature and will scale them such that the distribution is now centred around 0, with a **standard** deviation of 1. Standardization is useful for data which has negative values. It arranges the data in normal distribution. It is more useful in classification than regression.

The code is as follows:

```
from sklearn.preprocessing import StandardScaler
Standard=StandardScaler()
x=Standard.fit_transform(x)
```

Step-3: Training and Testing:

Once the preprocessing step is completed the data is standardized then we go for the splitting of data into training data and testing data by using another module named **train_test_split which** is present in the sklearn.model_selection library

where **Scikit-learn** is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms and **Model_selection** is a method for setting a blueprint to analyse data and then using it to measure new data.

Selecting a proper model allows you to generate accurate results when making a prediction.
To do that, you need to train your model by using a specific data set. Then, you test the model against another data set.

train_test_split() is a function in **Sklearn model selection** for splitting data arrays into **two subsets**: for training data and for testing data. With this function, you don't need to divide the data set manually.

By default, Sklearn **train_test_split** will make random partitions for the two subsets. However, you can also specify a random state for the operation.

```
from sklearn.model_selection import train_test_split
train_x,test_x,train_y,test_y=train_test_split(x,y,test_size=0.2)
```

Here we divided the data in 80:20 ratios which is denoted at test_size. After splitting the data we have to train the data using fit() method with respect to the algorithm and predict the values for the test data using predict() method and from the predicted and test outcomes we will get the accuracy for the algorithm by using accuracy_score() from the metrics module.

The Metric module implements functions assessing prediction error for specific purposes. These metrics are detailed in sections on Classification metrics, Multilabel ranking metrics, Regression metrics and Clustering metrics. Finally, Dummy estimators are useful to get a baseline value of those metrics for random predictions.

Step-4: cross-validation:

Cross-validation is a statistical method used to estimate the skill of machine learning models. It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

There are different methods to do cross-validation. They are mentioned below:

1. K-fold cross-validation
2. Leave One Out (LOO)
3. Leave P Out (LPO)

In Cross-validation, the training set is split into *k* smaller sets. From the above three we will use K-fold cross-validation.

The following procedure is followed for each of the *k* “folds”:

- A model is trained using K−1 of the folds as training data;
- The resulting model is validated on the remaining part of the data (i.e., it is used as a test set to compute a performance measure such as accuracy).

The performance measure reported by *k*-fold cross-validation is then the average of the values computed in the loop. This approach can be computationally expensive, but does not waste too much data, which is a major advantage in problems such as inverse inference where the number of samples is very small.

CHAPTER 7

PSUEDO CODES

7.1 GRADIENT BOOSTING

Input:

- A training data set D = {(x1, y1), (x2, y2), ..., (xn, yn)}

- The number of iterations or boosting rounds T

- The maximum depth of each tree d

- The learning rate or step size alpha

- A loss function to optimize

Procedure Gradient_Boosting_Machine(D, T, d, alpha, loss_function):

1. Initialize a vector F to zeros with the same size as the number of training instances in D

2. For t = 1 to T do:

3. Compute the negative gradient of the loss function with respect to F, i.e. -(dL/dF)

4. Fit a regression tree to the negative gradient by recursively splitting the data into regions that minimize the loss function. The maximum depth of the tree is d.

5. Compute the step size by minimizing the loss function over the region of each leaf of the tree

6. Update the F vector by adding alpha times the step size times the predictions of the current tree to the previous F vector

7. Return the Gradient Boosting Machine model

7.2 EXTREME GRADIENT BOOSTING MODEL

Input:

Training set (X, y), Testing set (X_test, y_test), Number of rounds (n_rounds),

Learning rate (eta)

Procedure:

1. Initialize a model with base score as the mean of the training labels, and an empty list of trees.

2. For each round i in 1 to n_rounds:

   a. Compute the gradient and hessian of the loss function for the current predictions using the training set.

   b. Train a decision tree on the training set using the computed gradient and hessian.

   c. Compute the predicted values for the training set using the trained decision tree.

   d. Update the model by adding the trained tree with a learning rate of eta.

   e. Compute the predicted values for the testing set using the updated model.

   f. Calculate the testing error using a metric such as mean squared error or accuracy.

3. Return the trained model.

To predict the output for a new input x_new, follow these steps:

1. For each tree in the model, compute the predicted value using the features of x_new.

2. Combine the predicted values from all trees using the learning rate and the base score to get the final prediction for x_new.

7.3 LIGHT GRADIENT BOOSTING MODEL

Input:

Training set (X, y), Testing set (X_test, y_test), Number of trees (n_trees), Learning rate (eta), Maximum depth of tree (max_depth)

1. Convert the training set and testing set to LightGBM's internal data format.

2. Set the parameters for LGBM, including the number of trees, learning rate, and maximum depth of the tree.

3. Train the model using the training set and the parameters.

a. Initialize the model with a single leaf and the mean of the training labels as the initial prediction.

b. For each tree i in 1 to n_trees:

i. Compute the gradient and the hessian of the loss function for the current predictions using the training set.

ii. Train a decision tree on the training set using the computed gradient and hessian.

iii. Add the trained tree to the model using the leaf-wise growth strategy, which selects the leaf with the maximum reduction in the loss function as the next split.

iv. Compute the predicted values for the training set using the trained decision tree.

7.4 LOGISTIC REGRESSION

Input:

- A training dataset D = {(x1, y1), (x2, y2), ..., (xn, yn)}

- The learning rate or step size alpha

- The number of iterations or epochs

Procedure Logistic_Regression(D, alpha, T):

1. Initialize the weights w to zeros with the same size as the number of features in D

2. For t = 1 to T do:

3. Shuffle the training data D

4. For each training instance (x, y) in D do:

5. Compute the dot product of w and x, i.e. wx

6. Compute the predicted probability of y=1, i.e. p(y=1|x,w) = sigmoid(wx)

7. Compute the gradient of the loss function with respect to w, i.e. dL/dw = (p(y=1|x,w)-y) *x

8. Update the weights w by subtracting alpha times the gradient of the loss function from the previous w, i.e. w = w - alpha * (p(y=1|x,w)-y) * x

9. Return the Logistic Regression model

7.5 RANDOM FOREST

Input:

- A training dataset D = {(x1, y1), (x2, y2), ..., (xn, yn)}

- The number of trees in the forest T

- The number of features to consider at each split m

- A function to measure the quality of a split criterion.

Output:

- A Random Forest model consisting of T Decision Trees

Procedure Random_Forest(D, T, m, split_criterion):

For t = 1 to T do:

1. Draw a random sample of size n with replacement from D to create a new data set D_t
2. Randomly select m features from the total set of p features
3. Build a Decision Tree from the new data set D_t using the selected m features and the split_criterion
4. Store the Decision Tree in the Random Forest model

Return the Random Forest model

CHAPTER 8

IMPLEMENTATION CODE

8.1 Importing Libraries

```
import numpy as np

import sys

np.set_printoptions(threshold=10)

import matplotlib.pyplot as plt

import pandas as pd

%matplotlib inline
```

```
from google.colab import drive

drive.mount('/content/drive')
```

8.2 Loading the data set

```
#dataset=pd.read_csv('/content/drive/MyDrive/Ovarian_Cancer_Project-
main/Ovarian_Cancer_Project-main/ovariantotal.csv')

df=pd.read_csv('/content/drive/MyDrive/Ovarian_Cancer_Project-
main/Ovarian_Cancer_Project-main/ovariantotal.csv')

df.head()
```

8.3 Determining Mutual Information

```
mutual_info = pd.Series(mutual_info)
mutual_info.index = X_train.columns
mutual_info.sort_values(ascending=False)
mutual_info = pd.Series(mutual_info)

mutual_info.index = X_train.columns
```

```
mutual_info.sort_values(ascending=False)
```

8.4 Feature Scaling

```
from sklearn.preprocessing import StandardScaler

sc=StandardScaler()

X_train=sc.fit_transform(X_train)

X_test=sc.transform(X_test)



print(X_train)
```

8.5 Random Forests

```
import pandas as pd

import re

import matplotlib.pyplot as plt

import seaborn as sns

from scipy.stats import chi2_contingency



from sklearn.model_selection import train_test_split

from sklearn.model_selection import cross_val_score

from sklearn.model_selection import GridSearchCV

from sklearn.ensemble import RandomForestClassifier

from sklearn.metrics import accuracy_score, confusion_matrix, classification_report,precision
_score, f1_score, recall_score
```

```
%matplotlib inline


test_acc = accuracy_score(y_test, y_pred)

print(precision_score(y_test,y_pred))

print(recall_score(y_test, y_pred))

print(f1_score(y_test, y_pred))


print('Training confusion matrix')

print(confusion_matrix(y_train, y_t))

print('Testing confusion matrix')

print(confusion_matrix(y_test, y_pred))

print ("Train accuracy: {0:.4f}".format(train_acc))

print ("Test accuracy: {0:.4f}".format(test_acc))


'''print ("Feature Importance")


importances = m_best.feature_importances_

for i in importances:

    print(i)'''


pred_prob3 = m_best.predict_proba(X_test)

print(classification_report(y_test,y_pred))
```

8.6 Logistic Regression

```
from sklearn.model_selection import GridSearchCV

from sklearn.linear_model import LogisticRegression

from sklearn.metrics import accuracy_score, precision_score, f1_score, recall_score



grid_values = {'penalty': ['l1','l2'], 'C': [0.001,0.01,0.1,1,10,100,1000]}

lr=LogisticRegression(random_state=42)

model_lr = GridSearchCV(lr, param_grid=grid_values)



model_lr.fit(X_train,y_train)



m_best = model_lr.best_estimator_



y_t = m_best.predict(X_train)

train_acc = accuracy_score(y_train, y_t)



y_pred = m_best.predict(X_test)

print(y_pred)

test_acc = accuracy_score(y_test, y_pred)

print(precision_score(y_test,y_pred))

print(f1_score(y_test, y_pred))
```

```
print(recall_score(y_test,y_pred))

print(accuracy_score(y_test,y_pred))



print('Training confusion matrix')

print(confusion_matrix(y_train, y_t))

print('Testing confusion matrix')

print(confusion_matrix(y_test, y_pred))



pred_prob5 = m_best.predict_proba(X_test)

print(classification_report(y_test,y_pred))
```

8.7 Gradient Boosting Machine

```
baseline = GradientBoostingClassifier(learning_rate=0.1, n_estimators=100,max_depth=3, mi
n_samples_split=2, min_samples_leaf=1, subsample=1,max_features='sqrt', random_state=42)



baseline.fit(X_train, y_train)



gbm_predict_train = baseline.predict(X_train)



#get accuracy

gbm_accuracy = metrics.accuracy_score(y_train, gbm_predict_train)
```

```
#print accuracy
print ("GBM training Accuracy: {0:.4f}".format(gbm_accuracy))


gbm_predict_test = baseline.predict(X_test)


#get accuracy
gbm_accuracy_testdata = metrics.accuracy_score(y_test, gbm_predict_test)
#print accuracy
print ("GBM testing Accuracy: {0:.4f}".format(gbm_accuracy_testdata))
print(precision_score(y_test,y_pred))
print(f1_score(y_test, y_pred))
print(recall_score(y_test,y_pred))


from sklearn.metrics import log_loss
logloss = log_loss(y_test, gbm_predict_test)
print ("GBM Log Loss: {0:.4f}".format(logloss))


from sklearn.metrics import roc_auc_score
auc = roc_auc_score(y_test, gbm_predict_test)
print ("GBM AUC: {0:.4f}".format(auc))
```

```
#print ("Confusion Matrix for GBM")

# labels for set 1=True to upper left and 0 = False to lower right

#print ("{0}".format(metrics.confusion_matrix(y_test, gbm_predict_test, labels=[1, 0])))



#print ("")



print ("Classification Report\n")



# labels for set 1=True to upper left and 0 = False to lower right

print ("{0}".format(metrics.classification_report(y_test, gbm_predict_test, labels=[1, 0])))
```

8.8 Light Gradient Boosting Machine

```
d_train = lgb.Dataset(X_train, label=y_train)

d_test = lgb.Dataset(X_test, label=y_test)



watchlist = [d_train, d_test]



model = lgb.train(params, train_set=d_train, num_boost_round=1000, valid_sets=watchlist, ea
rly_stopping_rounds=50, verbose_eval=4)



y_t = model.predict(X_train)

y_t = np.where(y_t > 0.5, 1, 0)
```

```
train_acc = metrics.accuracy_score(y_train, y_t)


y_pred = model.predict(X_test)

y_pred = np.where(y_pred > 0.5, 1, 0)


test_acc = metrics.accuracy_score(y_test, y_pred)


precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

auc = roc_auc_score(y_test, y_pred)

logloss = log_loss(y_test, y_pred)


print ("Train accuracy: {0:.4f}".format(train_acc))

print ("Test accuracy: {0:.4f}".format(test_acc))

print ("Precision: {0:.4f}".format(precision))

print ("Recall: {0:.4f}".format(recall))

print ("F1: {0:.4f}".format(f1))

print ("AUC: {0:.4f}".format(auc))

print ("Log Loss: {0:.4f}".format(logloss))
```

```
print (metrics.classification_report(y_test, y_pred))
```

8.9 Extreme Gradient Boosting Machine

```
grid_search.fit(X_train, y_train)



grid_search.best_estimator_



m_best = grid_search.best_estimator_

y_t = m_best.predict(X_train)

train_acc = accuracy_score(y_train, y_t)



y_pred = m_best.predict(X_test)



test_acc = accuracy_score(y_test, y_pred)

precision = precision_score(y_test, y_pred)

recall = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

print(accuracy_score(y_test,y_pred))



print ("Train accuracy: {0:.4f}".format(train_acc))

print ("Test accuracy: {0:.4f}".format(test_acc))

print ("Precision: {0:.4f}".format(precision))
```

```
print ("Recall: {0:.4f}".format(recall))

print ("F1: {0:.4f}".format(f1))



print('Training confusion matrix')

print(confusion_matrix(y_train, y_t))

print('Testing confusion matrix')

print(confusion_matrix(y_test, y_pred))



pred_prob2 = m_best.predict_proba(X_test)



cfm=confusion_matrix(y_test,y_pred)

print(cfm)



print("Classification Report: ")

print(classification_report(y_test,y_pred)
```

8.10 Voting Classifier

```
estimator = []

estimator.append(('LR', LogisticRegression(random_state=1)))

estimator.append(('GBC', GradientBoostingClassifier(learning_rate=0.1, n_estimators=100,m
ax_depth=3, min_samples_split=2, min_samples_leaf=1, subsample=1,max_features='sqrt',

 random_state=42)))
```

```
estimator.append(('XGB', XGBClassifier()))

estimator.append(('LGBM', lgb.LGBMClassifier()))

#estimator.append(('ADB', AdaBoostClassifier()))

estimator.append(('RF', RandomForestClassifier(n_estimators = 1000, random_state = 42)))


from sklearn.ensemble import VotingClassifier

vot_hard = VotingClassifier(estimators = estimator, voting ='hard')

vot_hard.fit(X_train, y_train)

y_pred = vot_hard.predict(X_test)


print(metrics.accuracy_score(y_test,y_pred))

print(metrics.precision_score(y_test,y_pred))

print(metrics.f1_score(y_test, y_pred))

print(metrics.recall_score(y_test,y_pred))
```

CHAPTER 9
MACHINE LEARNING LIBRARIES

Programming Language Used : Python

Below are the list of libraries.

9.1. SCIKIT-LEARN(SKLEARN)

Scikit-learn is both, well-documented and simple to learn/use.If one wants an introduction to machine learning, or to use ML testing tool, SCIKIT-LEARN lets programmer to construct a predictive data model with a few lines of code and then apply that model to the data as a high-level library. It is flexible and integrates nicely with other Python libraries such as Matplotlib for charts, Numpy for numerical computations, and Pandas for Data Frames.

Scikit-learn contains many supervised & unsupervised learning algorithms. It provides a selection of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.Most importantly, it is the simplest and cleanest ML library. Its primary API architecture is focused on being straightforward to use while remaining adaptable and flexible for research. Because of its robustness, it is suitable for use in any end-to-end ML project — from research to production deployments. It is based on the machine learning libraries mentioned below:

NumPy: is a Python library that allows you to manipulate multidimensional arrays and matrices. It also includes a large set of mathematical functions for performing various calculations.

SciPy: is an environment of libraries for performing technical programming tasks.

Matplotlib: is a library that can be used to build different charts and graphs.

9.1.1 NUMPY:

NumPyis a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. NumPy has incorporated features of the competing Numarray into Numeric, with extensive modifications. NumPy is open-source software and has many contributors.

Features:

NumPy targets the Python reference implementation of Python, which is a non-optimizing byte code interpreter. Mathematical algorithms written for this version of Python often run much slower than compiled equivalents. NumPy addresses the slowness problem partly by providing multidimensional arrays and functions and operators that operate efficiently on arrays, requiring rewriting some code, mostly inner loops using NumPy.

Python bindings of the widely used computer vision library OpenCV utilize NumPy arrays to store and operate on data. Since images with multiple channels are simply represented as three-dimensional arrays, indexing, slicing or masking with other arrays are very efficient ways to access specific pixels of an image. The NumPy array as universal data structure in OpenCV for images, extracted feature points, filter kernels and many more vastly simplifies the programming workflow and debugging.ms.

9.1.2 SCIPY

SciPy is a very popular library, it contains different modules for optimization, linear algebra, integration and statistics. The SciPy is one of the core packages that make up the SciPy stack. SciPy has various modules for implementing multiple Machine Learning algorithms.

9.1.3 MATPLOT

Matplotlib is a Python library which is defined as a multi-platform data visualization library built on Numpy array. It can be used in python scripts and other graphical user interface toolkit.

One of the greatest benefits of visualization is that it allows us visual access to huge amounts of data in easily digestible visuals. Matplotlib consists of several plots like line, bar, scatter, histogram etc. Several toolkits are available which extend Matplotlib functionality.

9.2 PANDAS:

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. In 2008, developer Wes McKinney started developing pandas when in need of high performance, flexible tool for analysis of data. Prior to Pandas, Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data — load, prepare, manipulate, model, and analyze. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc

Key Features of Pandas:

- Fast and efficient DataFrame object with default and customized indexing.
- Tools for loading data into in-memory data objects from different file formats.
- Data alignment and integrated handling of missing data
- Label-based slicing, indexing and subsetting of large data sets.
- High performance merging and joining of data..

9.3 SYS

The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter. The sys module gives the information such as What version of Python is running, The path to the Python interpreter that executes the current script, The command line parameters used when running the script etc.

LIST OF MODULES:

a) Sklearn.tree

b) Sklearn.model_selection

c) Sklearn.metrics

d) Sklearn.ensemble

e) Sklearn.linear_model

CHAPTER 10

RESULTS

As per our research, computing time required to train a model increases as more and more features are included. So, by the feature selection process, we have not considered the features that have little or no impact on the prediction of ovarian cancer. These characteristics may be crucial for predicting other malignancies but not in ovarian cancer.

Since only the best features were taken into account through k best features function, the predictive power of the model has not dropped. Moreover, fewer attributes aid in speeding up model computation. While voting classifier demonstrated accuracy of 90% for 15 features, random forest demonstrated accuracy of 91%.The three metrics by which the random forest model outperforms all other models are precision, recall and F1 score.

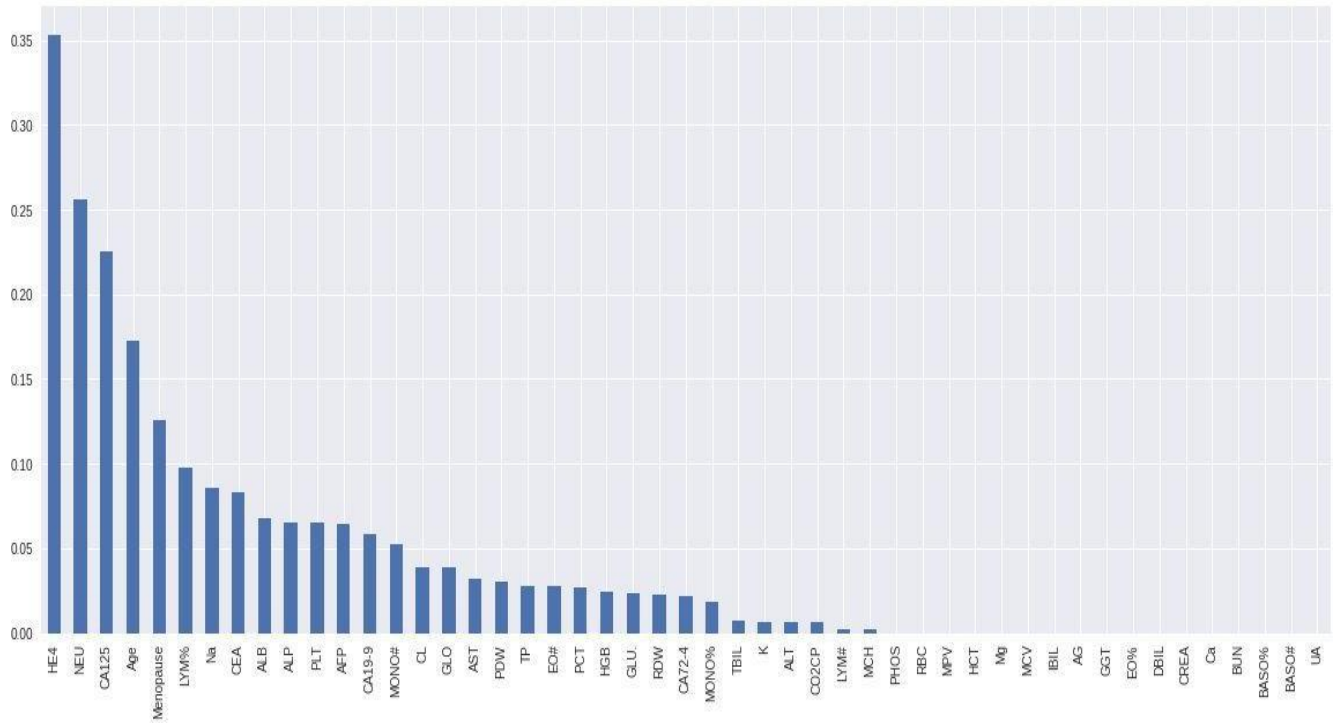

Figure 10.1: Mutual Information Values

For k=49 (number of features)

a) Random Forest

```
0.875
0.9210526315789473
0.8974358974358975
Training confusion matrix
[[135   4]
 [  0 140]]
Testing confusion matrix
[[27  5]
 [ 3 35]]
Train accuracy: 0.9857
Test accuracy: 0.8857
              precision    recall  f1-score   support

           0       0.90      0.84      0.87        32
           1       0.88      0.92      0.90        38

    accuracy                           0.89        70
   macro avg       0.89      0.88      0.88        70
weighted avg       0.89      0.89      0.89        70
```

b) Logistic Regression

```
0.8611111111111112
0.8157894736842105
0.8378378378378377
0.8285714285714286
Training confusion matrix
[[124  15]
 [  7 133]]
Testing confusion matrix
[[27  5]
 [ 7 31]]
              precision    recall  f1-score   support

           0       0.79      0.84      0.82        32
           1       0.86      0.82      0.84        38

    accuracy                           0.83        70
   macro avg       0.83      0.83      0.83        70
weighted avg       0.83      0.83      0.83        70
```

c) Gradient Boosting Machine

```
GBM training Accuracy: 1.0000
GBM testing Accuracy: 0.8714
0.8947368421052632
0.8947368421052632
0.8947368421052632
GBM Log Loss: 4.6342
GBM AUC: 0.8717
Classification Report
```

	precision	recall	f1-score	support
1	0.89	0.87	0.88	38
0	0.85	0.88	0.86	32
accuracy			0.87	70
macro avg	0.87	0.87	0.87	70
weighted avg	0.87	0.87	0.87	70

d) Light Gradient Boosting Machine

```
Train accuracy: 0.9211
Test accuracy: 0.8857
Precision: 0.8750
Recall: 0.9211
F1: 0.8974
AUC: 0.8824
Log Loss: 4.1193
```

	precision	recall	f1-score	support
0	0.90	0.84	0.87	32
1	0.88	0.92	0.90	38
accuracy			0.89	70
macro avg	0.89	0.88	0.88	70
weighted avg	0.89	0.89	0.89	70

e) Extreme Gradient Boosting Machine

```
0.020828705
0.0
0.019277811
0.026112475
0.0
0.015617629
0.0
0.027856821
0.0
[[28  4]
 [ 4 34]]
Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.88      0.88        32
           1       0.89      0.89      0.89        38

    accuracy                           0.89        70
   macro avg       0.88      0.88      0.88        70
weighted avg       0.89      0.89      0.89        70
```

f) Voting Classifier

```
0.8857142857142857
0.8947368421052632
0.8947368421052632
```

	precision	recall	f1-score	support
0	0.88	0.88	0.88	32
1	0.89	0.89	0.89	38
accuracy			0.89	70
macro avg	0.88	0.88	0.88	70
weighted avg	0.89	0.89	0.89	70

e) Extreme Gradient Boosting Machine

```
Train accuracy: 1.0000
Test accuracy: 0.9000
0.9
Precision: 0.9189
Recall: 0.8947
F1: 0.9067
AUC: 0.9005
Training confusion matrix
[[139   0]
 [  0 140]]
Testing confusion matrix
[[29  3]
 [ 4 34]]
[[29  3]
 [ 4 34]]
Classification Report:
```

	precision	recall	f1-score	support
0	0.88	0.91	0.89	32
1	0.92	0.89	0.91	38
accuracy			0.90	70
macro avg	0.90	0.90	0.90	70
weighted avg	0.90	0.90	0.90	70

f) Voting Classifier

	precision	recall	f1-score	support
0	0.88	0.88	0.88	32
1	0.89	0.89	0.89	38
accuracy			0.89	70
macro avg	0.88	0.88	0.88	70
weighted avg	0.89	0.89	0.89	70

For k=15 (number of features)

a) Random Forest

```
0.9473684210526315
0.9230769230769231
Training confusion matrix
[[123  16]
 [  1 139]]
Testing confusion matrix
[[28  4]
 [ 2 36]]
Train accuracy: 0.9391
Test accuracy: 0.9143
              precision    recall  f1-score   support

           0       0.93      0.88      0.90        32
           1       0.90      0.95      0.92        38

    accuracy                           0.91        70
   macro avg       0.92      0.91      0.91        70
weighted avg       0.92      0.91      0.91        70
```

b) Logistic Regression

```
0.8604651162790697
0.9135802469135803
0.9736842105263158
0.9
Training confusion matrix
[[117  22]
 [  4 136]]
Testing confusion matrix
[[26  6]
 [ 1 37]]
              precision    recall  f1-score   support

           0       0.96      0.81      0.88        32
           1       0.86      0.97      0.91        38

    accuracy                           0.90        70
   macro avg       0.91      0.89      0.90        70
weighted avg       0.91      0.90      0.90        70
```

c) Gradient Boosting Machine

```
GBM training Accuracy: 1.0000
GBM testing Accuracy: 0.8714
0.868421052631579
0.868421052631579
0.868421052631579
GBM Log Loss: 4.6342
GBM AUC: 0.8692
Classification Report
```

	precision	recall	f1-score	support
1	0.87	0.89	0.88	38
0	0.87	0.84	0.86	32
accuracy			0.87	70
macro avg	0.87	0.87	0.87	70
weighted avg	0.87	0.87	0.87	70

d) Light Gradient Boosting Machine

```
Train accuracy: 0.9247
Test accuracy: 0.9000
Precision: 0.8780
Recall: 0.9474
F1: 0.9114
AUC: 0.8956
Log Loss: 3.6044
```

	precision	recall	f1-score	support
0	0.93	0.84	0.89	32
1	0.88	0.95	0.91	38
accuracy			0.90	70
macro avg	0.90	0.90	0.90	70
weighted avg	0.90	0.90	0.90	70

e) Extreme Gradient Boosting Machine

```
0.8571428571428571
Train accuracy: 0.9713
Test accuracy: 0.8571
Precision: 0.8684
Recall: 0.8684
F1: 0.8684
Training confusion matrix
[[131   8]
 [  0 140]]
Testing confusion matrix
[[27  5]
 [ 5 33]]
[[27  5]
 [ 5 33]]
Classification Report:
              precision    recall  f1-score   support

           0       0.84      0.84      0.84        32
           1       0.87      0.87      0.87        38

    accuracy                           0.86        70
   macro avg       0.86      0.86      0.86        70
weighted avg       0.86      0.86      0.86        70
```

f) Voting Classifier

```
0.9
0.8974358974358975
0.9090909090909091
0.9210526315789473
```

```
              precision    recall  f1-score   support

           0       0.90      0.88      0.89        32
           1       0.90      0.92      0.91        38

    accuracy                           0.90        70
   macro avg       0.90      0.90      0.90        70
weighted avg       0.90      0.90      0.90        70
```

CHAPTER 11

CONCLUSION & FUTURE SCOPE

The early detection of ovarian cancer is crucial for improving patient outcomes, as the disease is often asymptomatic in its early stages and can be difficult to diagnose. In order to save computing time and improve model performance, relevant biomarkers must be identified using feature selection. These discovered biomarkers aid in the early detection of ovarian cancer.

Early stage detection of ovarian cancer is difficult due to lack of symptoms, low incidence rate, variations in tumor types etc. Our data set is limited to only 349 patients, if larger data set is available the models will be trained on more number of samples which might increase the performance of models. Our project could be developed in the future to distinguish between various ovarian cancer sub-types. The clinical data set was used to anticipate the incidence of ovarian cancer; moving forward, the project can be expanded to include image data set.

k=number of features	Model	Accuracy	Precision	Recall	F1_score
49	RF	0.88	0.87	0.92	0.89
	LR	0.82	0.86	0.81	0.83
	XGB	0.88	0.89	0.89	0.89
	GBM	0.87	0.89	0.89	0.89
	LGBM	0.88	0.87	0.92	0.89
	Voting	0.88	0.89	0.90	0.89
25	RF	0.90	0.89	0.92	0.90
	LR	0.84	0.82	0.89	0.86
	XGB	0.90	0.91	0.89	0.90
	GBM	0.85	0.82	0.89	0.86
	LGBM	0.87	0.85	0.92	0.88
	Voting	0.88	0.89	0.88	0.89
15	RF	0.91	0.90	0.94	0.92
	LR	0.90	0.86	0.97	0.91
	XGB	0.85	0.86	0.86	0.86
	GBM	0.87	0.86	0.86	0.86
	LGBM	0.90	0.87	0.94	0.91
	Voting	0.90	0.89	0.92	0.90

Table 11.1 Accuracy and evaluation metrics

REFERENCES

[1] Md. Martuza Ahamad, Sakifa Aktar, Md. Jamal Uddin, Tasina Rahman, Salem, Samer AI-Ashhab, AKM Azad, and Mohammad Ali Moni "Early Stage Detection Of Ovarian Cancer Based on Clinical Data Using Machine Learning Approaches", Journal Of Personalized Medicine, Vol. 12, July 2022.

[2] Anna F.Han and Patrick Emedom-Nnamdi "Predicting Ovarian Cancer Using Regularized Logistic Regression", July 2021.

[3] SuthamerthiElavarasu, Viji Vinod and Elavarasan Elangovan "Machine Learning Applications in Ovarian Cancer Prediction: A Review", International Journal of Pure and Applied Mathematics, Vol. 117, Issue 20, 2017.

[4] Xiaoyan Yang, Matlob Khushi and kamaran Shaukat, "Biomarker CA125 Feature Engineering and Class Imbalance Learning Improves Ovarian Cancer Prediction", IEEE Access 2020.

[5] Munetoshi Akazawa and Kazunori Hashimoto, "Artificial Intelligence in Ovaian Cancer Diagnosis", Anticancer Research 40, 4795-4800, 2020.

[6] Sreeja Sarojini, Ayala Tamir, Heeiin Lim, Shihona Li, Shifana Zhang, Andre Goy, Andrew Pecora and Stephen "Early Detection Biomarkers for Ovarian Cancer", Jounal Of Oncology, Volume 2012.

[7] Mingyang Lu, Zhenjiang Fan, Bin Xu, Lujun Chen, Xiao Zheng, Jundong Li, Qi Mi and Jingting Jiang "Using Machine Learning to predict Ovarian Cancer", Journal od Medical Informatics, 1386-5056, May 2020.

[8] Vincent Dochez, Helene Caillon, Edouard Vaucel, Jerome Dimet, Norbert Winer and Guillaume Ducarme "Biomarkers and algorithms for diagnosis of ovarian cancer: CA125, HE4, RMI and ROMA a review", Issue 10, 2019.

[9] Kokila. R. Kasture and Praven N.Matte, "Prediction and Classification of Ovarian Cancer using Enhanced Deep Convolutional Layer", Volume 70, Issue 3, 310-318, March 2022.

[10] Mansi Mathur, Vikas Jindal and Gitanjali Wadhwa, "Detecting Malignancy of Ovarian Tumour Using Convolutional Layer: A Review", IEEE Access, 2020.

Certificate of Presentation

International Conference on
Recent Advances in Science, Technology, Engineering and Management

ICRASTEM-2K23 on 11th April 2023

Organised By

Department of Computer Science and Engineering

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(AUTONOMOUS)

Permanently Affiliated to JNTU Kakinada, Approved by AICTE, Accredited by NAAC with 'A' Grade, ISO 9001:2015 Certified, All eligible B.Tech branches Accredited by NBA, Nambur, Pedakakani (M), Guntur (Dt.) – 522508.

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

Society for Learning Technologies SOLETE India

This is to certify that the paper entitled Ovarain Cancer Prediction in Early Stage Using Machine Learning Approaches with author(s) Dr. K.L. Lakshmi., P.H. Chandana., P.H. Sri., N.N. Kumar., N. Hemath was presented by – in the ICRASTEM - 2K23 International Conference on Recent Advances in Science, Technology, Engineering and Management, held on 11th April 2023 at Vasireddy Venkatadri Institute of Technology, Guntur Dt. in association with SOLETE (Society for Learning Technologies) Vijayawda.

Mr. T. Kranthi Kumar
CEO, SOLETE

Dr. V. Ramachandran
Convener
ICRASTEM-2K23
Professor & HOD CSE, VVIT

Dr. Y. Mallikarjuna Reddy
Principal, VVIT

IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

Ovarain Cancer Prediction in Early Stage Using Machine Learning Approaches

DOI:10.48047/IJFANS/V11/I12/186

Dr. K. Lohitha Lakshmi[1], Associate Professor, Department of CSE,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.
P. Hima Chandana[2], **P. Hema Sri**[3], **N. Nitish Kumar**[4], **N. Hemanth**[5]
[2,3,4,5]UG Students, Department of CSE,
Vasireddy Venkatadri Institute of Technology, Nambur, Guntur Dt., Andhra Pradesh.
lohita.kanchi@gmail.com[1], himachandanap@gmail.com[2],
hemaperumalla20@gmail.com[3], nitishkumarnavuluri@gmail.com[4],
nallamothuhemanth50@gmail.com[5]

Abstract

Ovarian cancer is a disorder of ovarian cell growth that is triggered by series of acquired mutations affecting a single cell or its clonal progeny. It is purposeless prey on host and virtually autonomous. It is usually diagnosed at a late stage because of poor sensitivity of screening test. There are still no effective cures for this illness. Still early detection might lower the mortality rate. Our project's major goal is to conduct predictive analytics for early detection by using machine learning models and statistical techniques on clinical data collected from specific patients. Mutual information testing is crucial in statistical analysis for identifying indicative biomarkers. A collection of machine learning models, such as the Random Forest (RF), Extreme Gradient Boosting Machine (XGBoost), Logistic Regression (LR), Gradient Boosting Machine (GBM), and Light Gradient Boosting Machine (LGBM)are utilized in the classificationof ovarian tumors as benign or malignant. By using proposed system, it can significantly identify the class of benign and malignant patients. The data collected is analyzed and pre-processed before it is used for model training and testing.

Keywords: Biomarkers, Mutual Information, Ovarian Cancer, Predictive Models, Risk Factors.

1. Introduction

Ovarian cancer is an abnormal mass of ovarian tissue whose growth outpaces and is not coordinated with the normal tissue[10]. It also continues to grow excessively even after the initial stimulus that caused the alteration has stopped. Although most of the cases are typically seen in elderly women, ovarian cancer is also being diagnosed in young women. Due to inadequate screening methods and a lack of distinctive symptoms, more than 70% of ovarian cancer patients receive a diagnosis of the disease in a late stage. Trans Vaginal Ultra Scan (TVS) and Cancer Antigen 125 (CA125) Blood Test are the current screening methods available to detect ovarian cancer. Usually elderly women present with pain abdominal distention. In some cases it is associated with bloating. Age, certain genetic mutations, and a family history of the disease are all risk factors for ovarian cancer. There are various studies evaluating the effectiveness of the biomarkers that distinguish between benign tumour and ovarian cancer. Researchers have been investigating the use of machine

Research paper © 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

learning algorithms to identify ovarian cancer which are derived from many data sources, including clinical data, genetic data also medical imaging, in recent years. The ability to predict disease progression and the diagnosis of cancer is a great potential of machine learning algorithms with new approaches. Machine learning is widely accepted technique to understand the prognosis of the disease. In order to develop predictive analytics for early detection, this study used mutual information to identify pertinent features and machine learning models.

2. Literature Survey

Martuza et al. made work to apply machine learning models to the clinical data which is considered from 349 patients[1]. 49 features are present which are sub divided into three groups. The results obtained from Random Forest(RF),Gradient Boosting Machine(GBM),Light Gradient Boosting Machine (LGBM) classifiers showed high level of accuracy of 88 percent.

Arcuda et al. employed serum proteome profile data to drive wavelet feature selection in machine learning methods.

Patrick et al. made efforts to predict ovarian cancer using regularized logistic regression[2]. 349 medical records from the "Third Affiliated Hospital of Soochow University" make up the data set used in this study. A logistic regression model is constructed with a Least Absolute Shrinkage and Selection Operator (LASSO) regularization penalty. Accuracy of 90.6% is obtained using Logistic regression.

SuthamerthiElavarasu et al. made a review on Machine Learning Applications in Ovarian Cancer Prediction[3]. Three principle approaches are stated for the selection of features, namely integrated, filters and envelopes approach. The survey concludes that by the analysis of various studies for predicting the outcomes of disease, the machine learning techniques and classification algorithms provides useful tools.

Viji Vinod et al. considered the TCGA ovarian cancer database of gene expression values. Machine learning model SVM showed the highest accuracy for recurrence and survival predictions.Yang et al. showed that the decision tree combined with Support Vector Machine-Synthetic Minority Over sampling Technique (SVMSMOTE) showed the top PPV (Positive Predictive Value) with 0.9041[4].

Munetoshi et al. had established that the histo-pathological identification of ovarian cancer may be predicted using artificial intelligence from preoperative assessment [5].

Lu et al. decided to evaluate 49 features from patient characteristics to build machine learning models. For performing modeling procedure they had used the combination of method named Minimum Redundancy Maximum Relevance (MRMR) feature selection, ReliefF feature selection also decision tree analysis. Finally they found that a prediction accuracy of 92.1% through decision tree approach using HE4 and carcino-embryonicantigen(CEA)[7].

Many studies have used CA125 as a marker of ovarian tumor[8]. To differentiate benign and malignant tumor In 1990, Jacobs et al. used the following factors- age, ultrasound status, clinical feature, menstrual history ,CA-125 levels as features to distinguish between benign and malignant ovarian tumors. Their experiment yielded a sensitivity of 81% and specificity of 75%.

He has used the data set consisting of 202 patients information from preoperative examinations. Highest accuracy of 80% was obtained using XGboost machine learning model.

R. Kasture et al. had used the histopathological images for the prediction of ovarian cancer[9]. Deep Learning method DCNN is used for training and evaluation[10]. Results of which highest accuracy of 91% was obtained from the KK-net model.

3. Problem Identification

Ovarian cancer is challenging to diagnose at an early stage because of non-specificity of the signs and symptoms. The problem with ovarian cancer is that it is often not recognized until it is advanced, making it more difficult to treat and less likely to have a successful out come. As the incidence of ovarian cancer is increasing day by day, early diagnosis can reduce the morbidity and mortality.

The significance of late-stage ovarian cancer diagnos is highlights the importance of early detection and the need for improved screening and diagnostic tools to improve patient outcomes[6]. Often there are challenges for developing treatment for all ovarian cancer patients such as heterogeneity of ovarian cancer and also side effects of the treatment. [12-20]

4. Methodology

Screening strategies that are available are Trans-vaginal Sonography(TVS) and CA125, but neither is specific enough to identify cancer when used alone. As a result we proposed a solution for early prediction of ovarian cancer. For early prediction identification of bio

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

markers plays a significant role. To identify the significant bio markers from the data set that helps in prediction we implemented a feature selection technique mutual info.

4.1 Data Set

The samples from patients with benign and malignant ovarian tumour used in this investigation were taken from a clinically tested raw data setwas gathered by the "Third Affiliated Hospital of Sooc how University" during the period of July 2017 to July 2018. A total of 349 patients samples are present in the present in the data set out of them there are 171 people with ovarian cancer and 178 people with benign tumours.

Figure 1 List of attributes[1]

4.2 Data Scaling

We have applied data scaling, often referred to as feature scaling, to transform the values of the many attributes in our data set on a single scale. Many of automated learning techniques are sensitive to the scale of input data and may not operate as intended if the data is not scaled properly. Data scaling is done by standardizing the data by subtracting the mean from actual value and dividing with the standard deviation.

$$\text{Scaled Value} = (\text{actual value} - \text{mean})/\text{deviation} \quad (1)$$

As a part of statistical analysis we have considered mutual information for feature selection.

4.3 Feature Selection

The process of selecting the pertinent characteristics from a huge data collection that contribute most to the prediction is known as feature selection[7].The fundamental objective of feature selection is to lessen model complexity and training time.

4.4 Mutual Information

A statistical entity known as mutual information measures the degree of dependency between two variables. In this context, mutual information is used to identify which

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

features or biomarkers are more strongly associated with the presence of ovarian cancer. Mutual information between two variables X and Y can be calculated as

$$MI(X,Y) = H(X)+H(Y)-H(X|Y) \quad (2)$$

Where H(X) is the entropy of X, H(Y) is the entropy of Y and H(X|Y) is the joint entropy.

$$H(X) = -\sum p(X) \log p(X) \quad (3)$$

$$H(Y) = -\sum p(Y) \log p(Y) \quad (4)$$

The Machine Learning models plays a significant role in prediction or classification. In this Study the Models implemented are Random Forest, Logistic Regression, Gradient Boosting Machine, Light Gradient Boosting Machine, Extreme Gradient Boosting and Voting Classifier.

5. Implementation

To implement the proposed methodology we have used python programming language. The popular python library Scikit Learn provides wide range of algorithms for data scaling, feature selection and model training.

5.1 Random Forest:

An ensemble learning technique, random forests or random decision forests build a large number of decision trees during the training phase. It is applied to both classification and regression models. Each decision tree in this model is created by selecting a subset of features and data points. A majority vote or average is used to evaluate the output.

5.2 Logistic Regression:

A statistical technique known as logistic regression is primarily used to predict the connection between a binary dependent variable and one or more independent variables for binary classification[2]. Sigmoid Function was calculated for the prediction of output class label.

5.3 Gradient Boosting Machine:

It is a boosting method which combines several weak predictors to form a strong predictor by optimizing a loss function. A Loss Function is calculated as the difference in predicted value and actual value. Each decision tree is constructed so as to minimize this error.

5.4 Light Gradient Boosting Machine(LGBM):

It is a framework designed to increase model efficiency and reduce memory usage. It works on the basis of two techniques - A. Gradient based one side sampling (GOSS). B. Exclusive

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

Feature bundling or EFB.GOSS is used to select the features that have high gradient which means features which has more predictive ability of class label where as EFB eliminates the features that are mutually exclusive.

5.5 Extreme Gradient Boosting(XGBoost):

It is an execution of gradient boosted decision tree in which weights plays an important role. Unlike all other boosting models XGboost has built parallel processing which makes it train the large datasets in less time.

5.6 Voting Classifier:

The ensemble voting classifier based its output prediction on the projected class with the highest probability after being trained on a variety of models.

Steps involved in the implementation:

1. Importing Library Functions
2. Loading of Data
   data = load_data()
3. Train and Test Split
   80% of the data is used for training and 20 % for testing.
4. Datascaling is done by the process of Standardization.
5. Feature Selection using mutual info.
6. Defining all the models.
7. Training the models with the features obtained.

The Evaluation metrics considered are:

Accuracy: The percentage of accurate predictions among all other predictions is known as accuracy.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad \textbf{(5)}$$

Precision: when a positive outcome was anticipated, precision is the percentage of true positives.

$$\text{Precision} = TP / (TP + FP) \quad \textbf{(6)}$$

Recall: Among all positive instances in the data set the predictions that are predicted as true positive.

$$\text{Recall} = TP / (TP + FN) \quad \textbf{(7)}$$

Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

F1 Score: The harmonic mean of recall and precision can be used to get the F1 Score.

$$F1\ score = 2 * (Precision * Recall) / (Precision + Recall) \quad (8)$$

TP - True Positive,

TN - True Negatives,

FP - False Positives,

FN - False Negatives.

6. Analysis Of Result

As per our research, computing time required to train a model increases as more and more features are included. So, by the feature selection process, we have not considered the features that have little or no impact on the prediction of ovarian cancer. These characteristics may be crucial for predicting other malignancies but not in ovarian cancer. Since only the best features were taken into account through k best features function, the predictive power of the model has not dropped. Moreover, fewer attributes aid in speeding up model computation. While voting classifier demonstrated accuracy of 90% for 15 features, random forest demonstrated accuracy of 91%.The three metrics by which the random forest model outperforms all other models are precision, recall and F1 score.

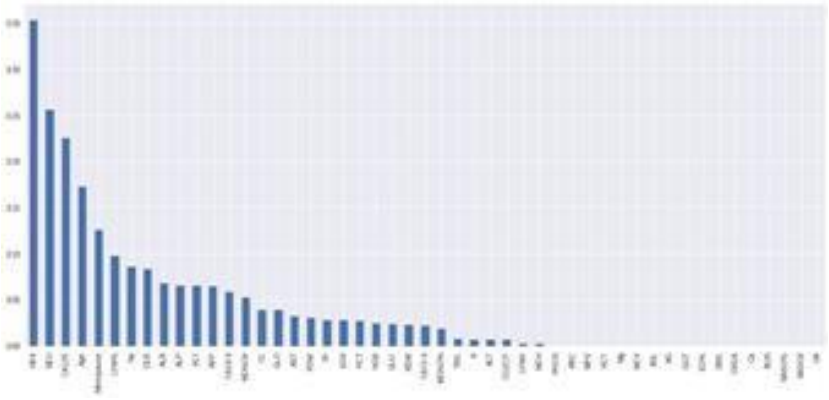

Figure 2: Mutual Information Values

Research paper © 2012 IJFANS. All Rights Reserved, UGC CARE Listed (Group -I) Journal Volume 11, Iss 12, Dec 2022

k=number of features	Model	Accuracy	Precision	Recall	F1_score
49	RF	0.88	0.87	0.92	0.89
	LR	0.82	0.86	0.81	0.83
	XGB	0.88	0.89	0.89	0.89
	GBM	0.87	0.89	0.89	0.89
	LGBM	0.88	0.87	0.92	0.89
	Voting	0.88	0.89	0.90	0.89
25	RF	0.90	0.89	0.92	0.90
	LR	0.84	0.82	0.89	0.86
	XGB	0.90	0.91	0.89	0.90
	GBM	0.85	0.82	0.89	0.86
	LGBM	0.87	0.85	0.92	0.88
	Voting	0.88	0.89	0.88	0.89
15	RF	0.91	0.90	0.94	0.92
	LR	0.90	0.86	0.97	0.91
	XGB	0.85	0.86	0.86	0.86
	GBM	0.87	0.86	0.86	0.86
	LGBM	0.90	0.87	0.94	0.91
	Voting	0.90	0.89	0.92	0.90

Table 1 Accuracy and evaluation metrics

7. Conclusion.

To conclude, in order to save computing time and improve model performance, relevant biomarkers must be identified using feature selection. These discovered biomarkers aid in the early detection of ovarian cancer.

8. Limitations & Future Scope

Early stage detection of ovarian cancer is difficult due to lack of symptoms, low incidence rate, variations in tumor types, accessibility of early stage detection test etc. Our data set is limited to only 349 patients, if larger data set is available the models will be trained on more number of samples which might increase the performance of models.

Our project could be developed in the future to distinguish between various ovarian cancer sub-types. The clinical data set was used to anticipate the incidence of ovarian cancer; moving forward, the project can be expanded to include image data set.

9. References

[1] Md. Martuza Ahamad, Sakifa Aktar, Md. Jamal Uddin, Tasina Rahman, Salem, Samer AI-Ashhab, AKM Azad, and Mohammad Ali Moni "Early Stage Detection Of Ovarian

Cancer Based on Clinical Data Using Machine Learning Approaches", Journal Of Personalized Medicine, Vol. 12, July 2022.

[2] Anna F.Han and Patrick Emedom-Nnamdi "Predicting Ovarian Cancer Using Regularized Logistic Regression", July 2021.

[3] SuthamerthiElavarasu, Viji Vinod and Elavarasan Elangovan "Machine Learning Applications in Ovarian Cancer Prediction: A Review", International Journal of Pure and Applied Mathematics, Vol. 117, Issue 20, 2017.

[4] Xiaoyan Yang, Matlob Khushi and kamaran Shaukat, "Biomarker CA125 Feature Engineering and Class Imbalance Learning Improves Ovarian Cancer Prediction", IEEE Access 2020.

[5] Munetoshi Akazawa and Kazunori Hashimoto, "Artificial Intelligence in Ovaian Cancer Diagnosis", Anticancer Research 40, 4795-4800, 2020.

[6] Sreeja Sarojini, Ayala Tamir, Heeiin Lim, Shihona Li, Shifana Zhang, Andre Goy, Andrew Pecora and Stephen "Early Detection Biomarkers for Ovarian Cancer", Jounal Of Oncology, Volume 2012.

[7] Mingyang Lu, Zhenjiang Fan, Bin Xu, Lujun Chen, Xiao Zheng, Jundong Li, Qi Mi and Jingting Jiang "Using Machine Learning to predict Ovarian Cancer", Journal od Medical Informatics, 1386-5056, May 2020.

[8] VincentDochez, Helene Caillon,Edouard Vaucel, Jerome Dimet, Norbert Winer and Guillaume Ducarme "Biomarkers and algorithms for diagnosis of ovarian cancer: CA125, HE4, RMI and ROMA a review", Issue 10, 2019.

[9] Kokila. R. Kasture and Praven N.Matte, "Prediction and Classification of Ovarian Cancer using Enhanced Deep Convolutional Layer", Volume 70, Issue 3, 310-318, March 2022.

[10] Mansi Mathur, Vikas Jindal and Gitanjali Wadhwa, "Detecting Malignancy of Ovarian Tumour Using Convolutional Layer: A Review", IEEE Access, 2020.

[11] Robert C.Bast, Zhen Lu, Chase Young Han, Karen H. Lu, Karen S. Anderson , Charles W. Drescher and Steven J. Stakes "Biomarkers and Strategies for Early Detection Of Ovarian Cancer", Departments of Experimental Therapeutics and Gynecologic Oncology, Texas University, October 2020.

[12] Sri Hari Nallamala, et al., "A Literature Survey on Data Mining Approach to Effectively Handle Cancer Treatment", (IJET) (UAE), ISSN: 2227 – 524X, Vol. 7, No 2.7, SI 7, Page No: 729 – 732, March 2018.

[13] Sri Hari Nallamala, et.al., "An Appraisal on Recurrent Pattern Analysis Algorithm from the Net Monitor Records", (IJET) (UAE), ISSN: 2227 – 524X, Vol. 7, No 2.7, SI 7, Page No: 542 – 545, March 2018.

[14] Sri Hari Nallamala, et.al, "Qualitative Metrics on Breast Cancer Diagnosis with Neuro Fuzzy Inference Systems", International Journal of Advanced Trends in Computer

Science and Engineering, (IJATCSE), ISSN (ONLINE): 2278 – 3091, Vol. 8 No. 2, Page No: 259 – 264, March / April 2019.

[15] Sri Hari Nallamala, et.al, "Breast Cancer Detection using Machine Learning Way", International Journal of Recent Technology and Engineering (IJRTE), ISSN: 2277-3878, Volume-8, Issue-2S3, Page No: 1402 – 1405, July 2019.

[16] Sri Hari Nallamala, et.al, "Pedagogy and Reduction of K-nn Algorithm for Filtering Samples in the Breast Cancer Treatment", International Journal of Scientific and Technology Research, (IJSTR), ISSN: 2277-8616, Vol. 8, Issue 11, Page No: 2168 – 2173, November 2019.

[17] Kolla Bhanu Prakash, Sri Hari Nallamala, et al., "Accurate Hand Gesture Recognition using CNN and RNN Approaches" International Journal of Advanced Trends in Computer Science and Engineering, 9(3), May – June 2020, 3216 – 3222.

[18] Sri Hari Nallamala, et al., "A Review on 'Applications, Early Successes & Challenges of Big Data in Modern Healthcare Management'", Vol.83, May - June 2020 ISSN: 0193-4120 Page No. 11117 – 11121.

[19] Nallamala, S.H., et al., "A Brief Analysis of Collaborative and Content Based Filtering Algorithms used in Recommender Systems", IOP Conference Series: Materials Science and Engineering, 2020, 981(2), 022008.

[20] Nallamala, S.H., Mishra, P., Koneru, S.V., "Breast cancer detection using machine learning approaches", International Journal of Recent Technology and Engineering, 2019, 7(5), pp. 478–481.