A
Project Report
On

Diagnosis of Thyroid Using Machine Learning

Submitted in partial fulfillment of the requirement for the award of
BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

UNDER THE GUIDANCE OF

MR. VAIBHAV KANT SINGH
(Assistant Professor)

SUBMITTED BY

Name of Students	University Roll No.
Priyanka Kumari	18103042
Rohan Gupta	18103047
Suraj Kumar	18103056

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING,

SCHOOL OF STUDIES OF ENGINEERING AND TECHNOLOGY,

GURU GHASIDAS VISHWAVIDYALAYA, CENTRAL UNIVERSITY,

BILASPUR, CHHATISGARH, INDIA

CERTIFICATE

We hereby certify that the work which is being presented in the Bachelor of Technology, Major Project Report entitled "Diagnosis of Thyroid using Machine Learning", in partial fulfilment of the requirements for the award of the **Bachelor of Technology in Computer Science and Engineering** and submitted to the Department of Computer Science and Engineering, School of Studies of Engineering and Technology, Guru Ghasidas Vishwavidyalaya (A Central University), Bilaspur, Chhattisgarh, India is an authentic record of our own work carried out during a period from December 2021 to April 2022 (8th semester) under the supervision of **Mr. Vaibhav Kant Singh,** Assistant Professor, Department of Computer Science & Engineering, SoS E&T, GGV, (Central University), Bilaspur, Chhattisgarh, India.

The matter presented in this Project Report has not been submitted by us or by anyone else for the award of any other degree elsewhere.

Signature of Students

PRIYANKA KUMARI	ROHAN GUPTA	SURAJ KUMAR
18103042	18103047	18103056

This is to certify that the above statement made by the students is correct to the best of my knowledge.

Signature of Supervisor

Mr. Vaibhav Kant Singh

(Assistant Professor)

Date:

Dr Alok Kumar Kushwaha

Head, Department of Computer Science & Engineering

DECLARATION

We here by declare that the work presented in this dissertation entitled **“DIAGNOSIS OF THYROID USING MACHINE LEARNING”** submitted to the **“Department of Computer Science & Engineering, School of Studies of Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, Chhattisgarh, India”.** Under the guidance of **Mr. Vaibhav Kant Singh**, Assistant Professor, Department of CSE, SoS E&T, GGV, Bilaspur, (C.G.), India has been done by us, and this report embodies our own work. The work is original as it has not been earlier submitted either in part or full for any purpose before by us or anyone else.

Name of Students

PRIYANKA KUMARI	ROHAN GUPTA	SURAJ KUMAR
18103042	18103047	18103056

ACKNOWLEDGEMENT

We express our sincere gratitude to **Mr. Vaibhav Kant Singh,** Assistant Professor, Department of Computer Science and Engineering, School of Studies of Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, Chhattisgarh, India, for his stimulating guidance, continuous encouragement and supervision throughout the course of present work.

We are extremely thankful to Associate Professor to **Dr. Alok Kumar Singh Kushwaha**, Associate Professor & Head Department of Computer Science and Engineering, School of Studies of Engineering and Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, Chhattisgarh, India for his generous guidance, help and useful suggestions.

We are extremely thankful to **Professor T.V. Arjunan**, Dean, School of Studies of Engineering & Technology, Guru Ghasidas Vishwavidyalaya, Central University, Bilaspur, Chhattisgarh, India for providing us infrastructural facilities to work in, without which this work would not have been possible.

Name of Students

PRIYANKA KUMARI	ROHAN GUPTA	SURAJ KUMAR
18103042	18103047	18103056

Date:

TABLE OF CONTENT

LIST OF FIGURES

ABSTRACT

The work is a continuation of the work done by the authors in the field of Medical Science. In the current scenario we are suffering from the problem of COVID. The pandemic is having a deep impact on various countries of the word. The pandemic has influence all across the world. There are various other diseases like cancer which is affecting a lot of population across the globe. In the current work the authors are engaged in the survey and exploration of ways to face a disease called Thyroid. In the current work the authors made survey of the Machine Learning approaches to make a detection of the Thyroid. In the current work the authors made a utilization of the Tool called Python. The authors made good utilization of the libraries present in Python. In the current work the authors surveyed and used the algorithms namely Gradient Boosting Classifier, ADA Boost Classifier, Light Gradient Boosting Machine, Decision Tree Classifier, Extra Tree Classifier, Logistic Regression, K-Neighbors Classifier, SVM-Linear Kernel, Linear Discriminant Analysis, Ridge Classifier, Dummy Classifier, Naïve Bayes and Quadratic Discriminant Analysis. The work done by the authors is an approach that still is having some limitations like the output user interface is not prepared which would have made the project more user interactive. The dataset is taken from Kaggle. The number of rows and columns present in the dataset of Kaggle is not that rich however the accuracies that are obtained after running of the code is extremely acceptable. But it would be more efficient if the data set would have considered more parameters and the number of tuples in the dataset would be more.

INTRODUCTION

Our thyroid gland produces hormones that play a role in many different systems of our body. When our thyroid gland produces these important hormones in excess or too little, it is called thyroid disease. There are various types of thyroid diseases such as hyperthyroidism, hypothyroidism, thyroiditis, and Hashimoto's thyroiditis.

Thyroid disorder is a popular time period for a scientific situation that continues our thyroid from making the proper quantity of hormones. our thyroid normally makes hormones that maintain our frame functioning normally. When the thyroid makes an excessive amount of thyroid hormone, your frame makes use of electricity too speedy. This is referred to as hyperthyroidism. Using electricity too speedy will do greater than make us tired — it could make our coronary heart beat faster, motive we to shed pounds without attempting or even make we experience nervous. On the flip-aspect of this, our thyroid could make too little thyroid hormone. This is referred to as hypothyroidism. When you've got too little thyroid hormone in our frame, it could make you experience tired, you would possibly benefit weight and you can also be not able to tolerate bloodless temperatures. These fundamental issues may be because of plenty of conditions. They also can be exceeded down thru families (inherited).

Who is affected by thyroid disease?

Thyroid disorder can have an effect on anyone including men, women, infants, young adults and the elderly. It may be gift at birth (normally hypothyroidism) and it could increase as you age (frequently after menopause in women). Thyroid disorder may be very common, with an envisioned 20 million humans withinside the Unites States having a few kind of thyroid disorder. A girl is set 5 to 8 instances much more likely to be recognized with a thyroid circumstance than a man.

Risk Factor

- Have a family history of thyroid disease.

- Have a medical condition (these can include pernicious anemia, type1 diabetes, primary adrenal insufficiency, lupus, rheumatoid arthritis, Sjogren’s syndrome and Turner syndrome).
- Take a medication that’s high in iodine(amiodarone).
- Are older than 60, especially in women.
- Have had treatment for a past thyroid condition or cancer (thyroidectomy or radiation).

Type of thyroid disease:

There are two main type of thyroid disease that are:

- Hypothyroidism
- Hyperthyroidism

Condition that can cause hypothyroidism include:

- **Thyroiditis:** This condition is inflammation (swelling) of the thyroid gland. Thyroiditis can reduce the amount of hormones produced by the thyroid gland.
- **Hashimoto’s thyroiditis:** Hashimoto's thyroiditis, which is painless, is an autoimmune disease in which cells of the body attack and damage the thyroid gland. This is an inherited state.
- **Postpartum thyroiditis**: This condition occurs in 5% to 9% of women after childbirth. It’s usually a temporary condition.
- **Iodine deficiency**: Iodine is used by the thyroid to produce hormones. An iodine deficiency is an issue that affects several million people around the world.
- **A non-functioning thyroid gland**: Sometimes, the thyroid gland doesn’t work correctly from birth. This affects about 1 in 4,000 newborns. If left untreated, the child could have both physical and mental issues in the future. All newborns are given a screening blood test in the hospital to check their thyroid function.

Condition that can cause hyperthyroidism include:

- **Graves’s disease:** In this condition the entire thyroid gland might be overactive and produce too much hormone. This problem is also called diffuse toxic goiter.

- **Nodules:** Hyperthyroidism can be caused by nodules that are overactive within the thyroid. A single nodule is called toxic autonomously functioning thyroid nodule, while a gland with several nodules is called a toxic multi-nodular goiter.

- **Thyroiditis:** This disorder can be either painful or not felt at all. In thyroiditis, the thyroid releases hormones that were stored there. This can last for a few weeks or months.

- **Excessive iodine:** When you have too much iodine (the mineral that is used to make thyroid hormones) in your body, the thyroid makes more thyroid hormones than it needs. Excessive iodine can be found in some medications (amiodarone, a heart medication) and cough syrups.

Common Symptoms:

Symptoms of hyperthyroidism can include:

- Experiencing anxiety, irritability and nervousness.
- Having trouble sleeping.
- Losing weight
- Having an enlarged thyroid gland or a goiter.
- Having muscle weakness and tremors.
- Experiencing irregular menstrual periods or having your menstrual cycle stop.
- Feeling sensitive to heat.
- Having vision problems or eye irritation.

Symptoms of an hypothyroidism can include:

- Feeling tired.
- Gaining weight.
- Experiencing forgetfulness.
- Having frequent and heavy menstrual periods.
- Having dry and coarse hair.
- Having a hoarse voice.
- Experiencing an intolerance to cold temperatures.

TOP 10 DEADIEST DISEASES IN INDIA

Here are the 10 Deadiest Diseases in india.

- **Cardiovascular Diseases:** Cardiovascular diseases are a range of conditions that affect your heart. They are the leading cause of deaths in India. Lifestyle risk factors, socio-economic changes, etc. are major causes of the rise of CVD.
- **Stroke:** A stroke occurs when the artery in your brain leaks or gets blocked. The symptoms of stroke include sudden numbness and confusion. It also causes vision loss and weakness. Read below in detail the symptoms, causes and preventive measures of Stroke.
- **Respiratory Diseases:** Respiratory infections including lung abscess, acute bronchitis and pneumonia are another biggest cause of death in India. It is one of the most common infections which affect adults.
- **Tuberculosis (TB):** It is an infectious disease that generally affects the lungs but may affect other body parts as well. But the good news is that tuberculosis is curable and preventable.
- **Chronic Obstructive Pulmonary Disease:** Chronic Obstructive Pulmonary Disease or COPD is a long-term lung disease that causes the patients difficulty in breathing. Not only in India, but COPD is responsible for taking the lives of many across the world.
- **Diabetes:** Diabetes affects insulin production and use. There are two types of Diabetes-Type 1 where the pancreas does not produce enough insulin and Type 2 where enough insulin is nor produced or it cannot be used effectively. Diabetes is a life-threatening disease.
- **Alzheimer’s Disease and other Dementias:** With Alzheimer’s disease comes not only loss of memory, but also loss of life in many cases. The progressive disease destroys memory and interrupts in activities like thinking, reasoning, etc.

- **Malaria:** Malaria is a fatal disease which is caused by Plasmodium parasite transmission by mosquitoes. It usually affects people in tropical and subtropical climates where parasites live.

- **Diarrheal Diseases:** Diarrhea is when you pass three or more loose stools in a day. It reduces the water and salt levels from your body making it weak. If it continues for days, then you may face dehydration.

- **Malignant and other Tumours:** Malignant tumours are cancerous and develop when cells grow without any control. It can grow to other parts of the body and spread as well making it life-threatening. The person may feel a tumour while often it is detected via imaging tests like MRI.

PROBLEM STATEMENT

In this Ever-Changing world, the wave of modernization is bringing a new wave of diseases. With the rising prices of goods, the prices of appointments with the medical fraternity are witnessing an exponential rise. In our daily lives, we can observe so many types of pollution around us. But are unaware of their damage to our daily lives. One strong observation can be drawn from the staggering 70,000 annual cases of Thyroid in India. In most cases, the disease tends to be asymptomatic in the early stages, making it nearly impossible to detect. To reduce the cost of screening, We need to develop an algorithm that can able to predict whether a patient is suffering from Thyroid or not depending upon the patient's health conditions.

As thyroid disorder are on the rise in india. Every People meet doctors to know that they are suffering from thyroid or not. Prediction of thyroid by doctor is a tedious process which might lead to negative prediction. Only experienced doctor can examine the case properly.

Sometimes, Human mind can do mistake to predict the thyroid but thyroid decteing by machine learning approach will not do mistake because they behave as we want. To assist doctor machine learning approach can help them in diagnosis of disease and reduce the burden of doctor.

Meachine learning approach will helpful for doctors as well as people. For doctors it will decrease the burden of doctors and for people at the initial stage of thyroid they can check at home that he/she suffering from thyroid or not.

LITERATURE SURVEY

S. No.	Title of the Paper	Authors	Remark
1.	Detection of Thyroid Using Machine Learning Approach	V.K. Singh, N.D. Yadav, R.K. Singh and M. Sahu	In this paper the author used ML approach for detection of thyroid
2.	Support Vector Machine based Diagnostic system for thyroid cancer using statistical texture features.	B Gopinath, N Shanthi	In the paper written by Gopinath and Shanthi 96.7% accuracy, 95% sensitivity and 100% specificity is observed. The wavelength taken is 4 and 45 is the angle of observation. The Final Results of diagnosis in FNAC images observed for thyroid cancer performed an effective work making a utilization of texture of statistical data. In the Paper the derived information regarding Gabor filters which are having SVM association gave effective result.
3.	High Accuracy Thyroid Tumor Image Recognition Based on Hybrid Multiple Models Optimization.	Wanrong Gu, Yijun Mao, Yichen He, Zaoqing Liang, Xianfen Xie, Ziye Zhang, And Weijiang Fan	This paper was published in July 10, 2020. In this article, As the main research object take the ultrasound image of Thyroid nodule. On the base of real world ultrasound image, result of this experiment showed that our proposed approach out performed the other method in accuracy and

			stability.
4.	Rainforest A framework for fast decision tree construction of large datasets.	Johannes Gehrke, Raghu Ramakrishnan, Venkatesh Ganti	In this article, we have developed a comprehensive approach for scaling classification trees. An algorithm that can be applied to all classification tree algorithms we know. The most important finding is the observation that the classification trees in the literature justify their division. A relatively compact, reference in the tree node of the AVC group for that node. The best division criteria developed in statistics and machine learning can now be used for large classifications.
5.	Research and Application of AdaBoost Algorithm Based on SVM.	Yanqiu Zhang, Ming Ni, Chengwu Zhang, Rujjie Li, Shang Liang, Sheng Fang, Zhouyu Tan	This paper suggest an SVM algorithm and AdaBoost algorithm that uses SVM as a weak Classifier to convert weak Classifier.
6.	Logistic Regression and Random Forest for Effective Imbalanced Classification.	Hanwa Luo, Xiubao Pan Qingshun Wang, Shasha Ye, Ying Qian	In this paper, the performance of random forest and logistic regression are compare on the prediction of imbalanced dataset. We used several ways to enhance two models based on cost sensitive learning to provide good accurate prediction when we are dealing with

			imbalanced datasets.
7.	Rainforest: A random forest algorithm for quantitative precipitation estimation over Switzerland.	Daniel Wolfensberger, Marco Gabella, Marco Boscacci, Urs Germann, and Alexis Berne	This paper is published in 29 April 2021. This Paper proposes a new database-driven QPE method. Switzerland can generate real-time 2D estimates of precipitation intensity on a 1km2 grid every 5 minutes. We approach this classic problem as follows: A new twist by training Random Forest (RF) regression Learn the QPE model directly from a large database spanning four years of observations that combines level and polarized radar. This algorithm has been carefully refined by optimization It uses its hyperparameters to compare with currently operational unpolarized QPE method. Ratings clearly show that this is possible with the HF algorithm Reduce the error and bias of predicted precipitation Intensity, especially for large, solid or mixed precipitation
8.	Diagnosis Method of Thyroid Disease Combining Knowledge Graph and Deep Learning.	Xuqing Chai	In the paper written by XUQING CHAI proposes diagnosis by connecting trivial and scattered knowledge spread across a variety of medical systems. The -

			--- of data available can be used for intelligent diagnosis of diseases.
9.	Liver Patient Classification using Logistic Regression.	Syed Hasan Adil, Mansoor Ebrahim, Kamran Raza, Syed Saad Azhar Ali, Manzoor Ahmed Hashmani	In this paper, a comprehensive and structural analysis is made on “Indian Liver Patient Records dataset published in UCI machine learning repository with a classification of accuracy of 74% with logistic Regression is achieved in the paper.
10.	A Comparison of Linear Discriminant Analysis and Ridge Classifier on Twitter Data.	Anagh Singh Shiva Prakash.B K.Chandrasekaran	In this paper Tikhonov’s Theory along with Ridge classifier is assessed. Utilization of Algorithm Levenberg Marquardt for classification is focused work on LDA which is essentially linear Discriminant Analysis is shown. Different aspect of the approaches discusses above are mentioned in the paper.
11.	Research on Algorithm of Decision Tree Induction.	Hua Ding Xiu-Kun Wang	In this paper the authors made a contribution on Decision Trees. In which entropy helps in avoiding the bugs of ID3. Some of the benefits of EMID if used are the simple structural of decision tree and higher degree in terms of reports generated on classification.

12.	A Systematic review on the role of artificial intelligence in sonographic diagnosis of thyroid cancer: Past, present and future	Fatemeh Abdolali, Michelle Noga, Atefeh Shahroudnejad, Abhilash Rakkunedeth, Hareendranathan, Jacob L Jaremko, Kumaradevan Punithakumar	In this jourals, there is review of CAD system for diagnosis of thyroid cancer with the help of machine learning and sonographic diagnosis. In this paper three different approached are used which are detection segmentation and as well as classification.
13.	High-Dimensional Quadratic Discriminant Analysis Under Spiked Covariance.	Houssem Sifaou, A.Kammoun, Mohsmed-Slim Alouini	This article suggested an improved QDA classifier Shows that it is superior to traditional RQDA while reducing computational complexity. The proposed classifier is better suited to a population with spikes in covariance. Situations commonly encountered in EEG signal processing, detection, and econometric applications. The results obtained are very promising and pave the way for extending the analysis to more general covariance models such as diagonal and low-ranked perturbations.
14.	Light Gradient Boosting machine for general sentiment classification on short texts: A comparative evaluation.	Fatima Alzamzami, Mohamd Hoda, Abdulmotaleb El Saddik	In this paper they used domain-free sentiment multimedia dataset to make a general multi-class sentiment classifier. On the basis of proven quality of the light

			gradient boosting machine, which handles high dimensional and disproportionate data. We trained LGBM modes to detect one of three tweet modes: positive, negative and neutral.
15.	Gradient Boosting Based classification of ion channels.	Divyansh Agrawal, Sachin Minocha, Suyel Nsmasudra, Sathish Kumar	This paper describes ion channel classification using gradient boosting algorithms such as light gradient boosting, extreme gradient boosting, and category boosting. The analysis of the gradient boosting algorithm and the comparison with the ANN classifier using various metrics shows the importance of work.
16.	Ai Meta-Learners and Extra Trees algorithm for the Detection of phishing websites.	Yazan Ahmad Alsariera, Victor Elijah Adeyemo, Abdullateef Oluwagbemiga Balogun, Ammar Kareem Alazzawi	The purpose of this research paper is to provide a good solution to the threat of phishing in modern society. Thus, this study aims to address existing shortcoming that already exist. The accuracy of this method is 98%.
17.	High Precision Error prediction algorithm based on ridge regression predictor for reversible data hiding.	Xingyuan Wang, Pengbo Liu	In this paper, ridge regression is based on high precision error prediction algorithm for lossless data hiding is proposed. Ridge regression is a least-squares algorithm with a penalty that

			solves the least squares overfitting problem. In this research data hiding and encryption algorithms play an important role in protecting information security.
18.	Classification with learning K-Nearest neighbours.	J. Laaksonen, E. Oja	The Nearest neighbour classifier, especially the KNN algorithm is one of the simplest but most efficient classification rules and is in fact widely used. Here are three fitting rules that you can use in iterative training for KNN classifiers. This is a new approach from the perspective of both statistical pattern recognition and supervised neural network learning.
19.	Learning without human expertise: A case study of the Double Bridge Problem.	Krzysztof Mossakowski, Jacek Mandziuk	This paper is written by Krzysztof Mossakowski and Jacek Mandziuk and published in February 2009. This paper, Estimate the number of tricks a pair of bridge players will take in the so-called Double Dummy Bridge Problem using an artificial neural network that is trained only in sample games and does not represent human knowledge or even the rules of the game.

20.	TDTD: Thyroid Disease Type Diagnostics.	Jamil Ahmed, M. Abdul Rehman Soomrani	In this paper MDC which stands for Medical Data Cleaning approach which made utilization of Bayesian isotonic regression algorithm to identify dependencies of thyroid disorder. This approach provided accuracy of around 95.7% when it is assessed on a cross validation of 10-k fold.
21.	Extreme Gradient Boosting Machine Learning Algorithm For Safe Auto Insurance Operations.	Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud	This paper written by Najmeddine Dhieb, Hakim Ghazzai, Hichem Besbes, and Yehia Massoud propose a framework for safe auto insurance operations which proposes the use of Extreme Gradient boosting for detecting fraudulent claims. It provides an added advantage for insurance companies by classifying different type and gives existing solution.
22.	An Implementation of Naïve Bayes Classifier.	Feng-Jen Yang	In the paper by Feng-Jen Yang titled "An Implementation of Naïve Byes Classifier", proposes the use of series of probabilistic computation to find the best-fitted classification model which here it root from Bayesian Theorem.

23.	Proposing Solution to XOR problem using minimum configuration MLP	V.K. Singh	Author Proposed an ANN Model
24.	Minimum Configuration MLP for Solving XOR problem	V.K. Singh and S. Pandey	Authors Proposed a Novel Model for ANN
25.	Proposing an Ex-NOR Solution using ANN	V.K. Singh and S. Pandey	Authors Proposed an Ex-NOR Model
26.	Mathematical Explanation To Solution For Ex-NOR Problem Using MLFFN	V.K. Singh	Mathematics behind ANN in LSP is conveyed by Author
27.	Mathematical Analysis for Training ANNs Using Basic Learning Algorithms	V.K. Singh	General Mathematics in ANN is portrayed
28.	Vector Space Model : An Information Retrieval System	V.K. Singh and V.K. Singh	An Information retrieval system is discussed in the article.
29.	Minimizing Space Time Complexity in Frequent Pattern Mining by Reducing Database Scanning and Using Pattern Growth Method	V.K. Singh and V Shah	Data Mining is discussed by the Authors
30.	The Huge Potential of Information Technology	V.K. Singh and V.K. Singh	Information Technology Landscape is discussed
31.	Proposing pattern growth methods for frequent pattern mining on account	V.K. Singh	Frequent Pattern Mining is discussed by the Author

	of its comparison made with the candidate generation and test approach for a given data set		
32.	RSTDB & Cache Conscious Techniques for Frequent Pattern Mining	V.K. Singh	RSTDB Algorithm is discussed by the Author
33.	Designing simulators for various VLSI designs using the proposed artificial neural network model TRIVENI	V.K. Singh	TRIVENI Model is discussed by Author
34.	Analysis of Stability and Convergence on Perceptron Convergence Algorithm	V.K. Singh	Convergence is given a look by the Author
35.	Machine Learning approach to detect Breast Cancer	V.K. Singh, A. Baghel, N.D. Yadav, M. Sahu and M. Jaiswal	Breast Cancer and Machine Learning Discussed by the authors
36.	SVM using rbf as kernel for Diagnosis of Breast Cancer	V.K. Singh	RBF Kernel is Discussed by the author
37.	Support Vector Machine using rbf, polynomial, linear and sigmoid as kernel to detect Diabetes Cases and to make a Comparative Analysis of the Models	V.K. Singh	Diabetes is Discussed by the Author

38.	Colorization of old gray scale images and videos using deep learning	V.K. Singh	Deep learning is used as Idea in the paper by the authors
39.	Dual Secured Data Transmission using Armstrong Number and Color Coding	V.K. Singh	Security aspect is discussed in the paper by the author
40.	Finding New Framework for Resolving Problems in Various Dimensions by the use of ES : An Efficient and Effective Computer Oriented Artificial Intelligence Approach	V.K. Singh, A. Baghel and S.K. Negi	Expert System is discussed by the Authors
41.	Twitter Sentiment Analysis	Chandrashekhar, R. Chauhan and V.K. Singh	The authors did ML Technology for Twitter Sentiment Analysis
42.	ML Approach for Detection of Lung Cancer	P. Kumari, R. Gupta, S. Kumar and V.K. Singh	ML is utilized in the area of Lung Cancer
43.	Automatic Number Plate Recognition	P. Sailokesh, S. Jupudi, I.K. Vamsi and V.K. Singh	Authors implemented Automatic Number Plate Recognition System
44.	Human Activity Recognition	Y.K. Reddy, K.M. Yadav and V.K. Singh	Authors Proposed Human Activity Recognition
45.	Text Summarization	R.N.R.K. Prasad, P.S.S.R Ram, S. Dinesh and V.K. Singh	Authors Proposed Text Summarization System
46.	Diagnosis of Breast Cancer using SVM taking polynomial as Kernel	V.K. Singh, N.D. Yadav and R.K. Singh	Detection of Breast Cancer is discussed

PROPOSED WORK

INTRODUCTION OF MACHINE LEARNING

Machine learning is an artificial intelligence (AI) application that allows you to automatically learn and improve from experience, even if the system is not explicitly programmed. Machine learning focuses on developing computer programs that can access data and use it for independent learning. The learning process begins with data such as observations or examples, first-hand experiences, instructions, etc., looks for patterns in the data, and makes better decisions in the future based on the examples provided. The main goal is for the computer to automatically learn and adjust actions accordingly, without human intervention or intervention. However, in traditional machine learning algorithms, text is displayed as a set of keywords. Instead, a semantic analysis-based approach mimics the human ability to understand the meaning of text. Machine learning algorithms are used in a variety of applications where it is difficult or impossible to develop traditional algorithms to perform the required tasks, such as medicine, email filtering, speech recognition, and computer vision. A subset of machine learning is closely related to computational statistics that focus on making predictions using computers. However, not all machine learning is statistical learning. Mathematical optimization research provides areas of methods, theories, and applications in the field of machine learning. Data mining is a related research area focused on exploratory data analysis by unsupervised learning. Some implementations of machine learning use data and neural networks in a way that mimics the functioning of the biological brain. In cross-business applications, machine learning is also known as predictive analytics.

History

The term machine learning was coined in 1959 by Arthur Samuel, an IBM employee in the United States and a pioneer in the field of computer games and artificial intelligence. Synonymous self learning computers were also used during this period. A typical book on machine learning research in the 1960s was Nilsson's Learning Machines, which was primarily related to machine learning for pattern classification. As explained by Duda and Hart in 1973, interest in pattern recognition continued until the 1970s. In 1981, a report was published on the use of educational strategies to learn that neural networks recognize 40 characters (26 characters, 10 digits, 4 special characters from computer terminals. Tom M. Mitchell provided a well-cited, more formal definition of algorithms studied in the field of machine learning. Performance on tasks is measured from T to P and improves with experience E. " This definition of a machine learning task does not cognitively define a field, but basically provides an operational definition, which is his paper" Computing Machinery and Intelligence "," Machinery. " The question "Can we think?" Is replaced by the question "Can machines do what we (as thinking beings) can do?" Modern machine learning has two goals. One is to classify the data based on the developed models, and the other is to predict future outcomes based on these models. Fictitious algorithms specific to data classification can be trained to classify cancerous bruises using computer vision of bruises in combination with supervised learning. Stock trading machine learning algorithms, on the other hand, can inform traders of potential future forecasts.

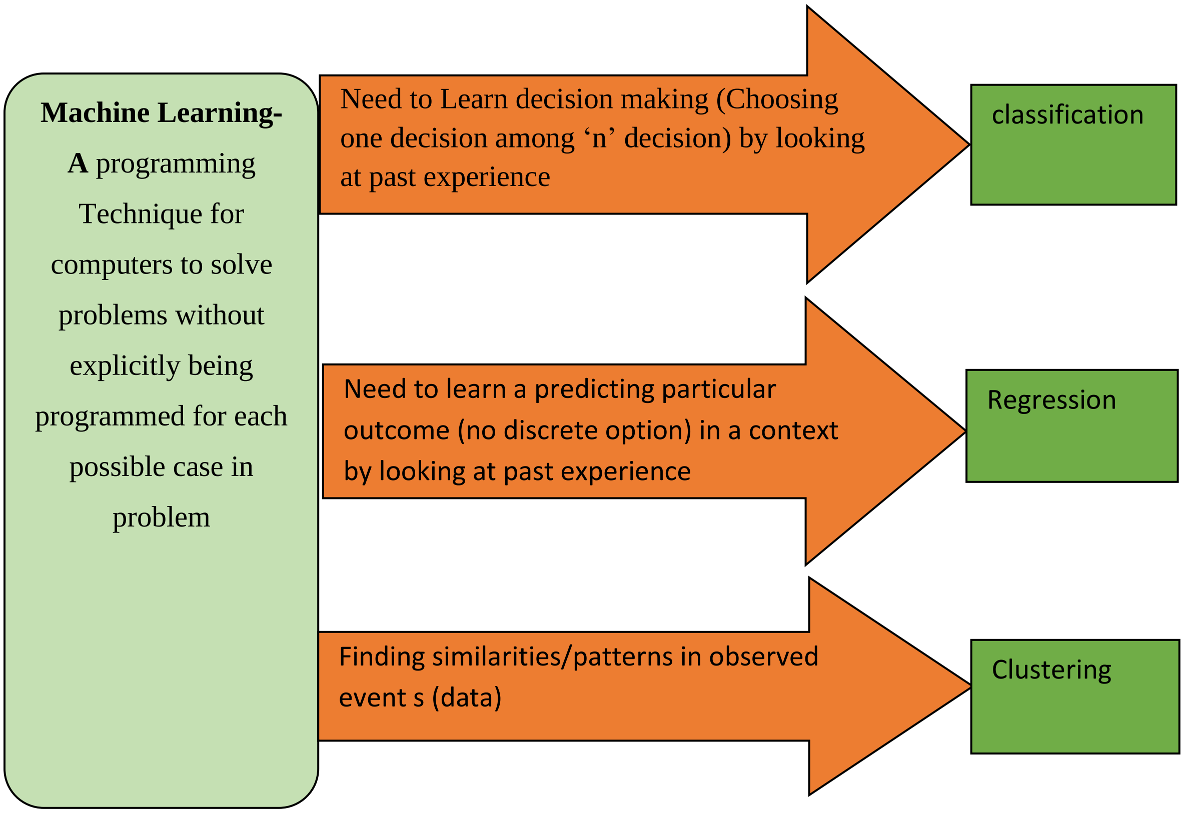

Machine Learning-
A programming Technique for computers to solve problems without explicitly being programmed for each possible case in problem
Need to Learn decision making (Choosing one decision among 'n' decision) by looking at past experience
classification
Need to learn a predicting particular outcome (no discrete option) in a context by looking at past experience
Regression
Finding similarities/patterns in observed event s (data)
Clustering

Fig 1: Machine Learning and its uses

Type of Machine Learning

Machine learning approaches are traditionally divided into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system:

- Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.
- Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data) or a means towards an end (feature learning).
- Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent). As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize.

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 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. Through iterative optimization of an objective function, supervised learning algorithms learn a function that can be used to predict the output associated with new inputs. An optimal function will allow the algorithm to correctly determine the output for inputs that were not a part of the training data. An algorithm that improves the accuracy of its outputs or predictions over time is said to have learned to perform that task.

Types of supervised learning algorithms include active learning, classification and regression. Classification algorithms are used when the outputs are restricted to a limited set of values, and regression algorithms are used when the outputs may have any numerical value within a range. As an example, for a classification algorithm that filters emails, the input would be an incoming email, and the output would be the name of the folder in which to file the email.

Similarity learning is an area of supervised machine learning closely related to regression and classification, but the goal is to learn from examples using a similarity function that measures how similar or related two objects are. It has applications in ranking, recommendation systems, visual identity tracking, face verification, and speaker verification.

Unsupervised learning

Unsupervised learning algorithms take a set of data that 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 labelled, 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.

Cluster analysis is the assignment of a set of observations into subsets (called clusters) so that observations within the same cluster are similar according to one or more predesignated criteria, while observations drawn from different clusters are dissimilar. Different clustering techniques make different assumptions on the structure of the data, often defined by some similarity metric and evaluated, for example, by internal compactness, or the similarity between members of the same cluster, and separation, the difference between clusters. Other methods are based on estimated density and graph connectivity.

Semi-supervised learning

Semi-supervised learning falls between unsupervised learning (without any labelled training data) and supervised learning (with completely labelled training data). Some of the training examples are missing training labels, yet many machine-learning researchers have found that unlabelled data, when used in conjunction with a small amount of labelled 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.

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, simulation-based 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.[39] Reinforcement learning algorithms do not assume knowledge of an exact mathematical model of the MDP, and are used when exact models are infeasible.

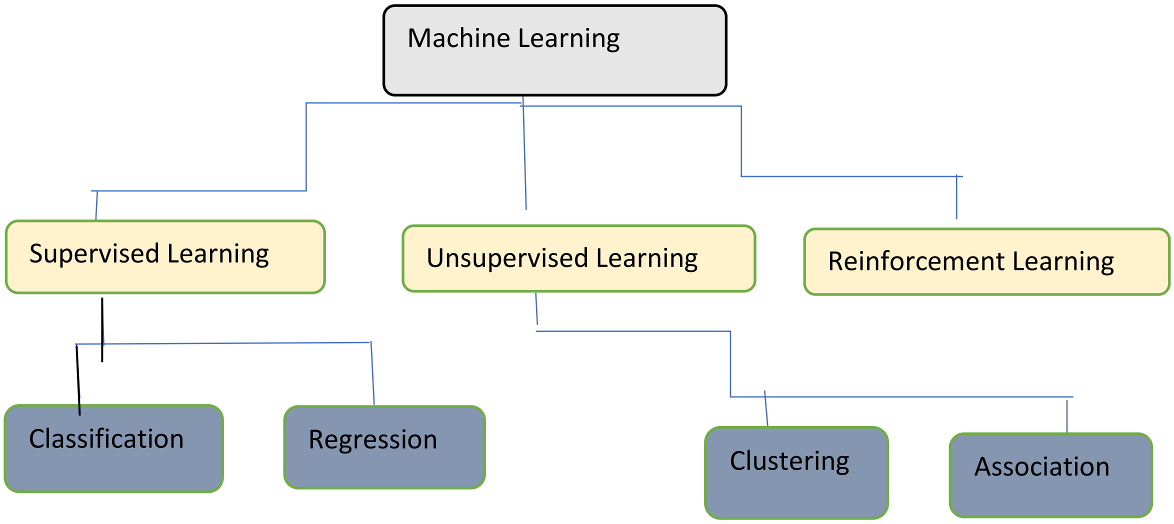


Fig 2: Machine learning and its type

1. Decision Tree Classifier-

A decision tree is a flowchart-like structure in which each internal node represents a feature test (e.g., whether a coin flip will land heads or tails), each leaf node represents a class label (decision made after computing all features), and branches represent feature combinations that lead to those class labels. The categorization rules are represented by the pathways from root to leaf.

In statistics, data mining, and machine learning, a decision tree is one of the predictive modeling approaches.

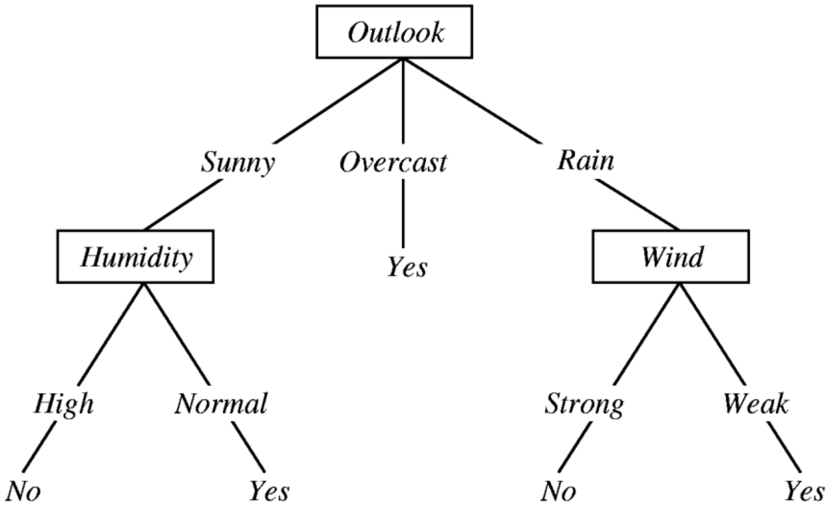


Fig 3: Decision Tree Classifier

2. Random Forest Classifier-

As the name suggests, a random forest is made up of a huge number of individual decision trees that work together as an ensemble. Each tree in the random forest generates a class prediction, and the class with the most votes becomes the prediction of our model.

The wisdom of crowds is the basic principle behind random forest, and it's a simple yet effective one. The trees defend each other from their unique flaws, which results in this magnificent effect. The random forest is a classification algorithm that uses numerous decision trees to classify data. When creating each individual tree, it employs bagging and feature randomization in order to generate an uncorrelated forest of trees whose committee prediction is more accurate than that of any one tree.

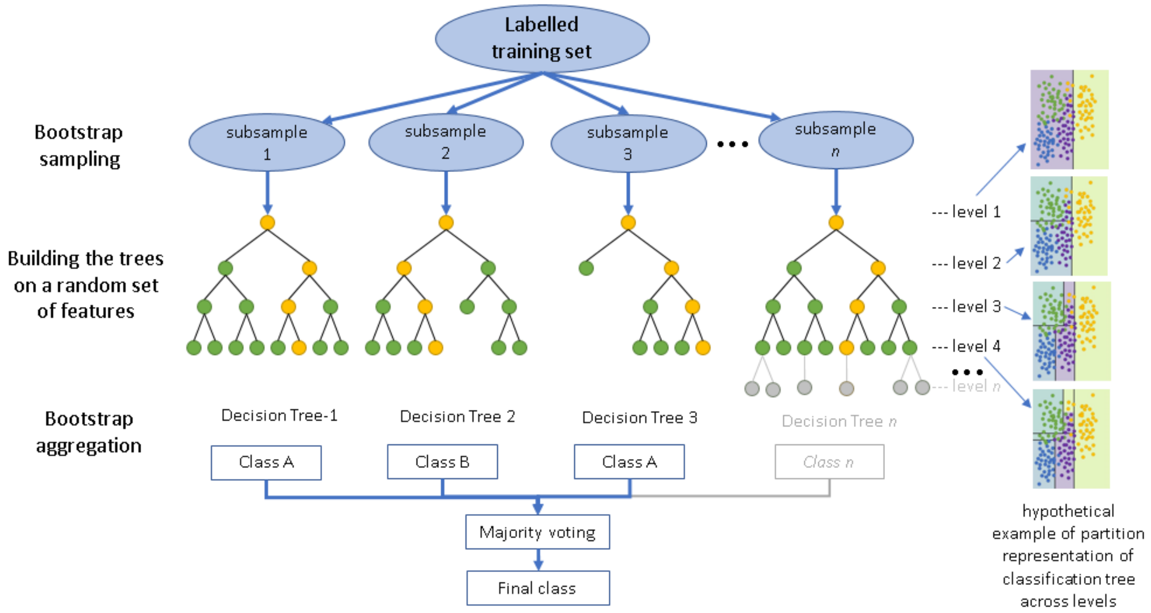


Fig:4 Random Forest Classifier-

3. Extra trees Classifier-

Extremely Randomized Trees Classifier (Extra Trees Classifier) is a kind of ensemble learning method that aggregates the results of multiple non-correlation decision trees collected in the "forest" and outputs the classification results. Conceptually, it's very similar to the Random Forest classifier, except that the decision tree is built in the forest.

The Extra Trees Forest's Decision Trees are all made from the original training sample. Then, at each test node, each tree is given a random sample of k features from the feature set, from which it must choose the best feature to split the data according to certain mathematical criteria (typically the Gini Index). Multiple de-correlated decision trees are created from this random sample of features.

During the construction of the forest, the normalised total reduction in the mathematical criteria used in the decision of feature of split (Gini Index if the Gini Index is used in the construction of the forest) is computed for each feature to perform feature selection using the above forest structure. The Gini Importance of the feature is the name given to this value. To execute feature selection, each feature is ranked in descending order by Gini Importance, and the user selects the top k features based on his or her preferences.

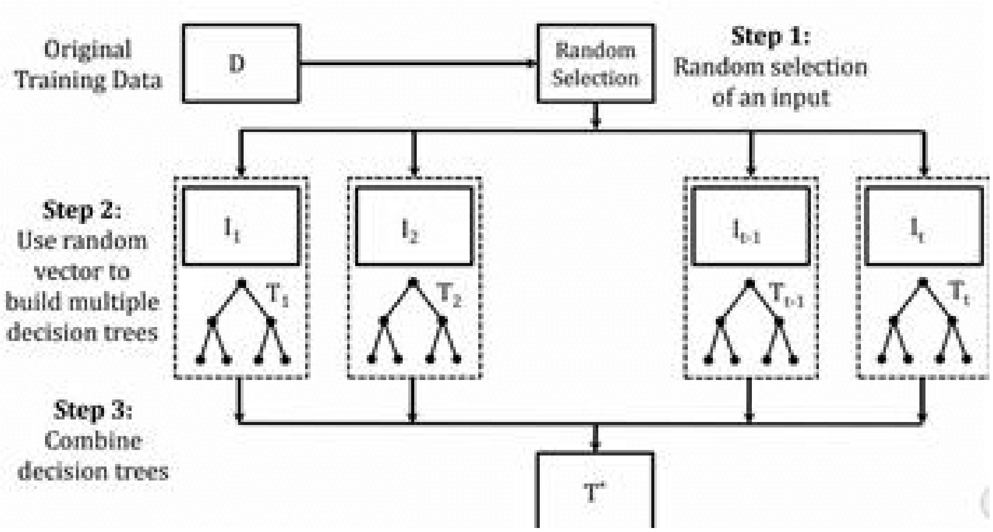

Original Training Data
D
Random Selection
Step 1: Random selection of an input
Step 2: Use random vector to build multiple decision trees
I_1
I_2
I_{t-1}
I_t
T_1
T_2
T_{t-1}
T_t
Step 3: Combine decision trees
T^*

Fig 5: Extra trees Classifier

4. Logistic Regression-

Logistic regression is a classification approach derived from statistics by machine learning. A statistical strategy for assessing a dataset in which one or more independent variables predict a result is known as logistic regression. The goal of logistic regression is to determine the model that best describes the connection between the dependent and independent variables.

Logistic regression is a machine learning classification technique. The dependent variable is modelled using a logistic function. The dependent variable is dichotomous, which means that only two classes are conceivable (eg.: either the cancer is malignant or not). As a result, while working with binary data, this strategy is applied.

logistic regression can be extended and further classified into three different types that are as mentioned below:

- **Binomial**: Where the target variable can have only two possible types. **eg**.: Predicting a mail as spam or not.
- **Multinomial**: Where the target variable have three or more possible types, which may not have any quantitative significance. **eg**.: Predicting disease.
- **Ordinal**: Where the target variables have ordered categories. **eg**.: Web Series ratings from 1 to 5.

- The sigmoid function is used in logistic regression to transfer predicted values to probabilities. This method converts any real value to a number between 0 and 1. At each point, this function has a non-negative derivative and exactly one inflection point.

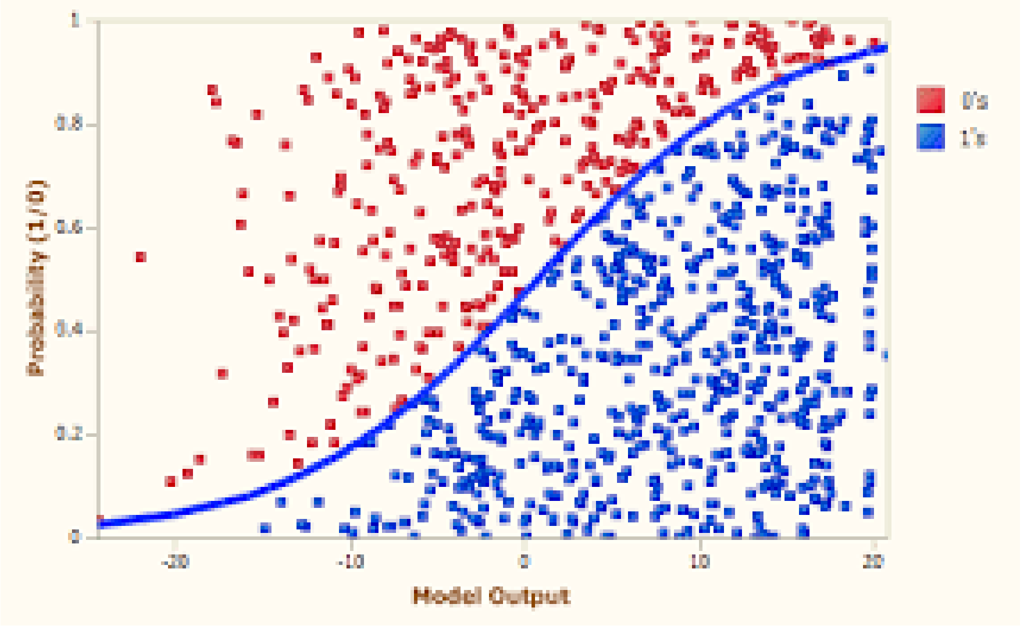


Fig 6: Logistic Regression

5. K Neighbors Classifier-

K-Nearest Neighbours is one of Machine Learning's most basic but crucial categorization algorithms. Pattern recognition, data mining, and intrusion detection are just a few of the applications it finds in the supervised learning domain.

It is commonly used in real-world contexts because it is non-parametric, which means it makes no underlying assumptions regarding data distribution (as opposed to other algorithms such as GMM, which assume a Gaussian distribution of the given data).

Prior data (also known as training data) is provided, which divides coordinates into groups based on an attribute.

The K-NN algorithm is a non-parametric algorithm, which means it makes no assumptions about the underlying data.

It's also known as a lazy learner algorithm since it doesn't learn from the training set right away; instead, it saves the dataset and performs an action on it when it comes time to classify it.

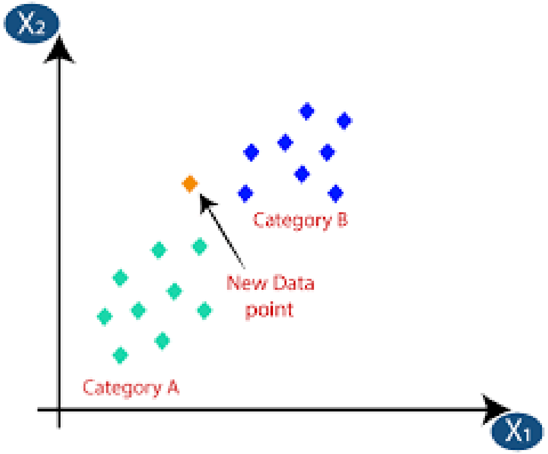


Fig 7: K Neighbors Classifier

6. **SVM - Linear Kernel-**

The Support Vector Machine is a supervised learning method that can be used for regression as well as classification. The key notion is that the algorithm tries to discover the best hyperplane based on the labelled data (training data) that can be used to categorise fresh data points. The hyperplane is a simple line in two dimensions.

Typically, a learning algorithm attempts to learn the most frequent characteristics (what distinguishes one class from another) of a class, and classification is based on those representative characteristics (so classification is based on differences between classes). The SVM operates in the opposite direction. It discovers the samples from different classes that are the most comparable. The support vectors will be those.

So other algorithms learn the differences while SVM learns similarities.

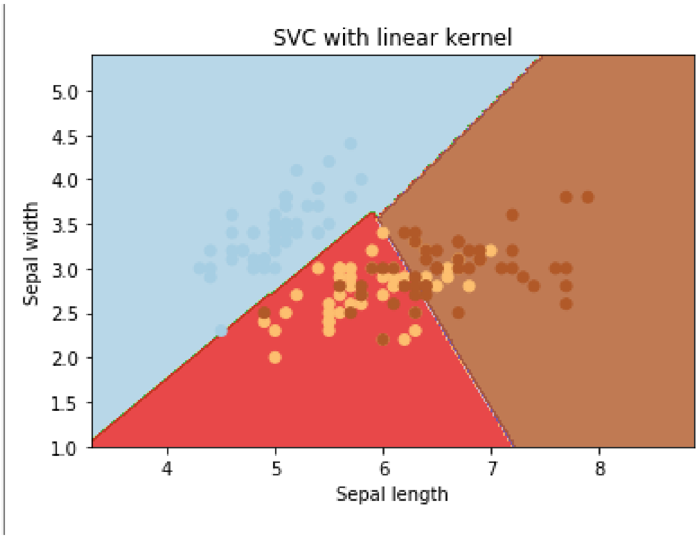


Fig 8: SVM - Linear Kernel

7. Linear Discriminant Analysis-

A dimensionality reduction technique known as Linear Discriminant Analysis, Normal Discriminant Analysis, or Discriminant Function Analysis is often employed for supervised classification problems. It's used to represent group differences, such as separating two or more classes. It is used to project higher-dimensional features onto a lower-dimensional space.

The classification algorithm logistic regression has typically been limited to two-class classification issues. Linear Discriminant Analysis is the recommended linear classification technique when there are more than two classes.

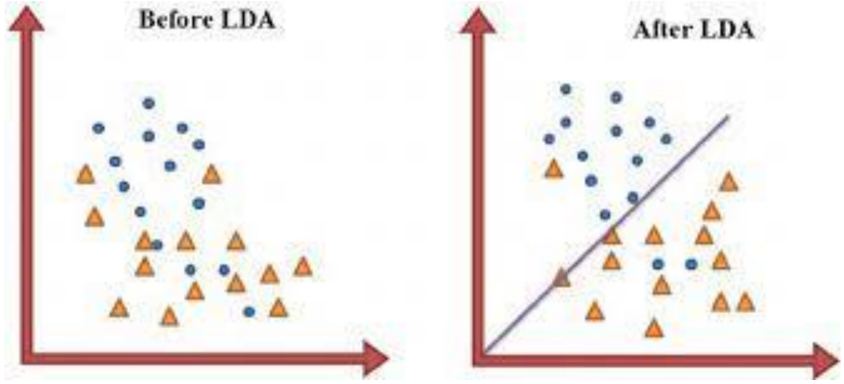


Fig 9: Linear Discriminant Analysis

8. **Ridge Regression-**

Ridge regression is a model tuning technique that can be used to analyse data with multicollinearity. L2 regularisation is achieved using this method. When there is a problem with multicollinearity, least-squares is unbiased, and variances are significant, resulting in projected values that are far from the actual values.

The first step in ridge regression is to normalise the variables (both dependent and independent) by dividing by their standard deviations and removing their means. This creates a notation problem because we need to declare whether the variables in a formula are standardised or not. All ridge regression computations are based on standardised variables in terms of standardisation. The final regression coefficients are rescaled to their original scale when they are displayed. The ridge trace, on the other hand, is on a standardised scale.

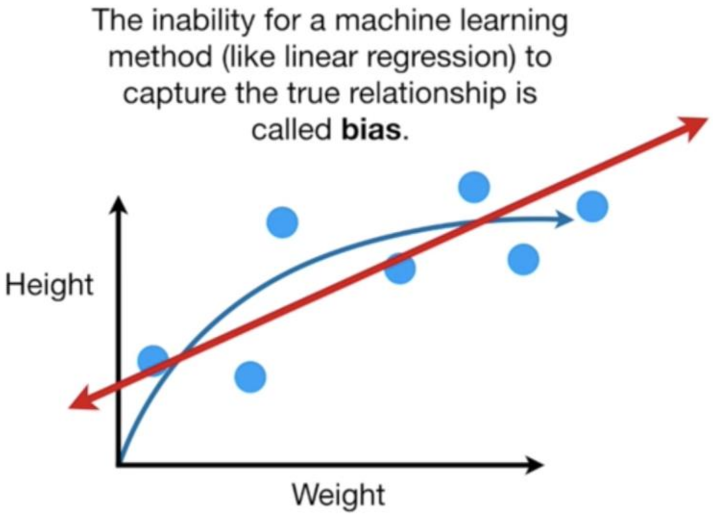

The inability for a machine learning method (like linear regression) to capture the true relationship is called **bias**.
Height
Weight

Fig 10: Ridge Regression

9. **Dummy Classifier-**

It's exactly what it sounds like: a dummy classifier! It's a type of classifier that produces predictions without looking for patterns in the data. The default model looks at the most common label in the training dataset and generates predictions based on that label. But, before we develop a dummy classifier, we must first understand how to compare the current model to the Dummy Classifier.

A dummy classifier is a sort of classifier that does not create any information about the data and instead classifies it according to simple principles. The classifier's behaviour is fully independent of the training data because the training data patterns are ignored and one of the techniques is used to predict the class label instead.

It serves just as a simple baseline for the other classifiers, with the expectation that any other classifier will outperform it on the supplied dataset. It's particularly effective for datasets with a known class imbalance. It is based on the belief that any analytic solution to a classification problem is preferable to a guessing method.

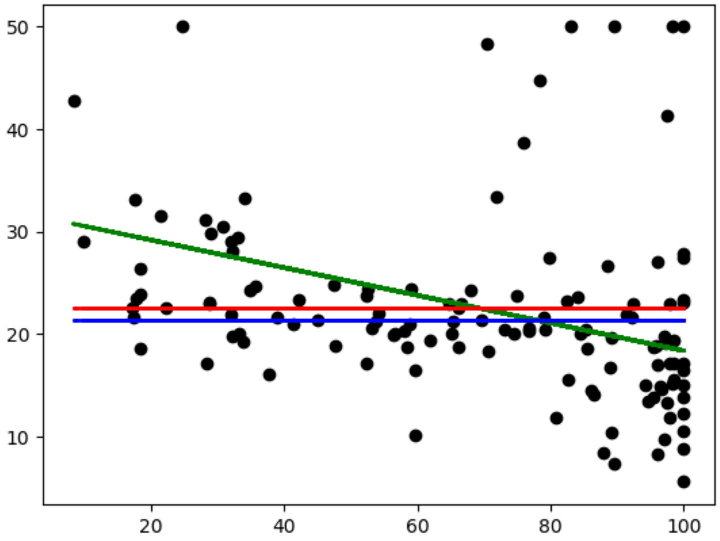


Fig 11: Dummy Classifier

10. Naive Bayes-

A Naive Bayes classifier is a probabilistic machine learning model that's used for classification task. The crux of the classifier is based on the Bayes theorem.

It is mostly utilised in text classification tasks that require a large training dataset.

The Nave Bayes Classifier is a simple and effective classification method that aids in the development of fast machine learning models capable of making quick predictions.

It's a probabilistic classifier, which means it makes predictions based on an object's probability.

Spam filtration, sentiment analysis, and article classification are all common uses of the Nave Bayes Algorithm.

It's termed Nave because it assumes that the appearance of one feature is unrelated to the appearance of other features. If the colour, shape, and flavour of the fruit are used to identify it, a red, spherical, and sweet fruit is identified as an apple. As a result, each aspect helps to identifying that it is an apple without relying on the others.

It's called Bayes since it's based on the Bayes' Theorem concept.

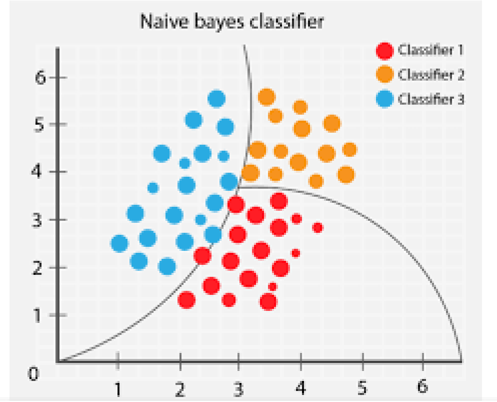


Fig 12: Naive Bayes

11. **Gradient Boosting Classifier-**

Each predictor in Gradient Boosting aims to improve on the previous one by lowering the mistakes. Gradient Boosting's unique concept is that, rather than fitting a predictor to the data at each iteration, it fits a new predictor to the residual errors created by the preceding prediction. Let's have a look at how Gradient Boosting Classification works in practise:

The method will obtain the log of the target feature's chances in order to make early predictions on the data. This is commonly calculated by dividing the number of True values (values of 1) by the number of False values (values equal to 0).

A popular boosting algorithm is gradient boosting. Each predictor in gradient boosting corrects the error of its predecessor. Unlike Adaboost, the training instance weights are not adjusted; instead, each predictor is trained using the predecessor's residual errors as labels.

CART is the base learner in a technique called Gradient Boosted Trees (Classification and Regression Trees).

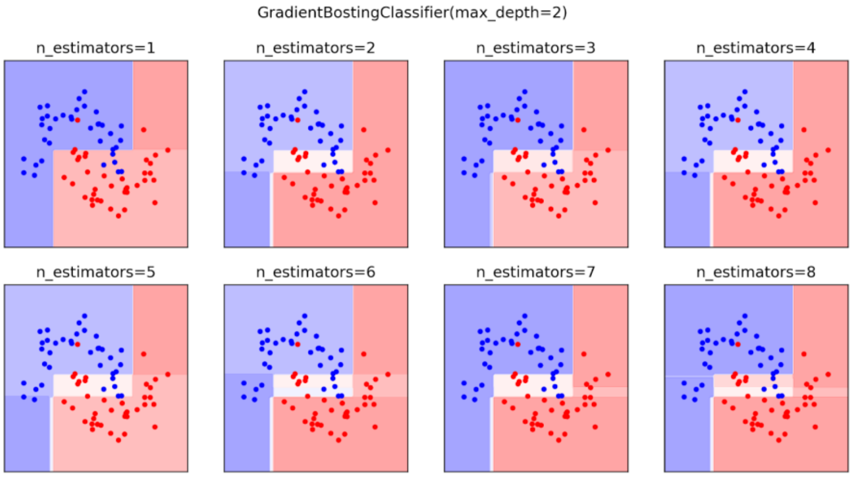


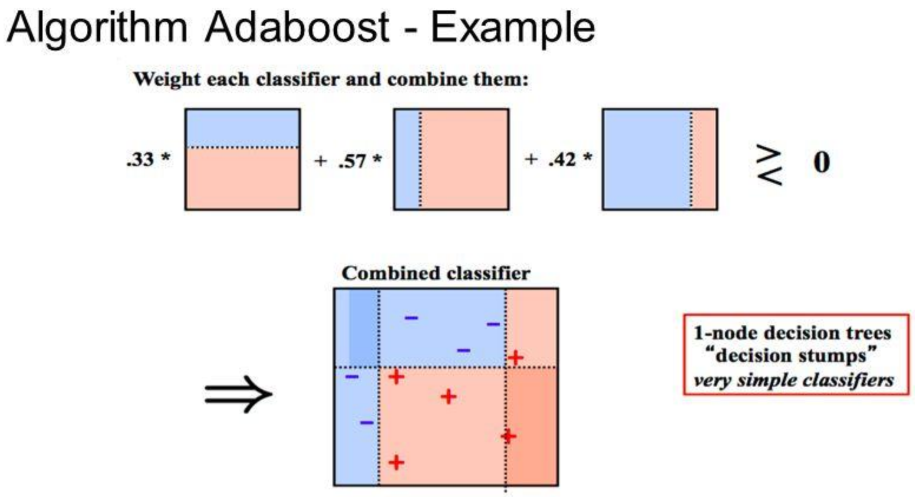
Fig 13: Gradient Boosting Classifier

12. **Ada Boost Classifier-**

The AdaBoost algorithm, short for Adaptive Boosting, is a Boosting approach used in Machine Learning as an Ensemble Method. The weights are re-allocated to each instance, with higher weights applied to improperly identified instances. This is termed Adaptive Boosting. In supervised learning, boost is used to reduce bias and variation. It is based on the notion of successive learning. Each subsequent student, with the exception of the first, is grown from previously grown learners. In other words, weak students are transformed into strong students. With a little modification, the AdaBoost method works on the same idea as boosting.

AdaBoost or Adaptive Boosting is one of the ensemble boosting classifier. It combines multiple weak classifiers to increase the accuracy of classifiers.

- AdaBoost is an iterative ensemble method. AdaBoost classifier builds a strong classifier by combining multiple poorly performing classifiers so that you will get high accuracy strong classifier.
- The basic concept behind Adaboost is to set the weights of classifiers and training the data sample in each iteration such that it ensures the accurate predictions of unusual observations.
- Any machine learning algorithm can be used as base classifier if it accepts weights on the training set.


Algorithm Adaboost - Example
Weight each classifier and combine them:
.33 *
+ .57 *
+ .42 *
≷ 0
Combined classifier
⇒
1-node decision trees
“decision stumps”
very simple classifiers

courtesy to Alexander Ihler http://sli.ics.uci.edu/Classes/2012F-273a?action=download&upname=10-ensembles.pdf

Fig 14: Ada Boost Classifier

13. Light gradient boosting machine

This is a gradient boosting framework that uses a tree-based learning algorithm, which is considered to be a very powerful algorithm for computation. This is considered a fast processing algorithm.

The tree of other algorithms grows horizontally, while the LightGBM algorithm grows vertically. That is, it grows leaf by leaf, and the other algorithms grow step by step. LightGBM chooses to grow with big losses. As the same hand grows, it can reduce more losses than a stepwise algorithm.

LightGBM is not for a small volume of datasets. It can easily overfit small data due to its sensitivity. It can be used for data having more than 10,000+ rows. There is no fixed threshold that helps in deciding the usage of LightGBM. It can be used for large volumes of data especially when one needs to achieve a high accuracy.

It has become difficult for the traditional algorithms to give results fast, as the size of the data is increasing rapidly day by day. LightGBM is called “Light” because of its computation power and giving results faster. It takes less memory to run and is able to deal with large amounts of data.

Most widely used algorithm in Hackathons because the motive of the algorithm is to get good accuracy of results and also brace GPU learning.

The size of the data grows rapidly from day to day, making it difficult for traditional algorithms to provide results quickly. LightGBM is called "Light" because of its processing power and faster delivery of results. It requires less RAM and can handle large amounts of data. The motivation for the

hackathon's most common algorithm is to improve the accuracy of the

results and support GPU tilt.

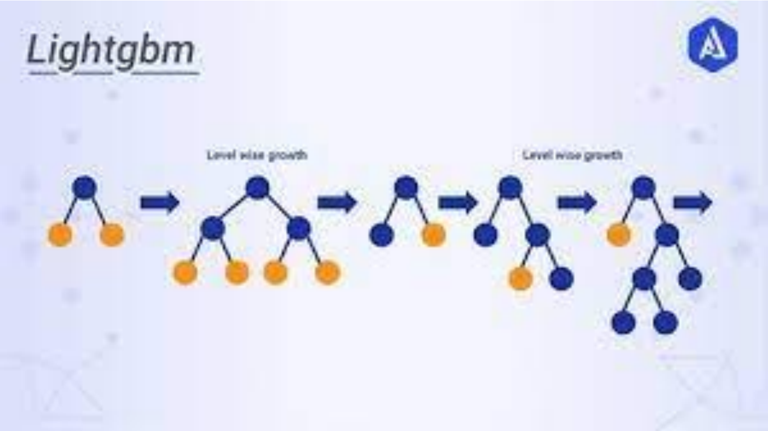


FIG 15: Light gradient boosting machine

14. **Quadratic discriminant analysis**

QDA is not much different from LDA, but assumes that the covariance matrix can be different for each class, so we estimate the covariance matrix individually for each class k, k = 1, 2, ..., K quadratic discriminant function:

This quadratic discriminant function is very much like the linear discriminant function except that because Σk, the covariance matrix, is not identical, you cannot throw away the quadratic terms. This discriminant function is a quadratic function and will contain second order terms. Classification rule: The classification rule is similar as well. You just find the class k which maximizes the quadratic discriminant function. The decision boundaries are quadratic equations in x. QDA, because it allows for more flexibility for the covariance matrix, tends to fit the data better than LDA, but then it has more parameters to estimate. The number of parameters increases significantly with QDA. Because, with QDA, you will have a separate covariance matrix for every class. If you have many classes and not so many sample points, this can be a problem. As we talked about at the beginning of this course, there are tradeoffs between fitting the training data well and having a simple model to work with. A simple model may fit the data as well as a complex model. Simple models don't fit training data like complex models, but they are more robust and may work better with test data.

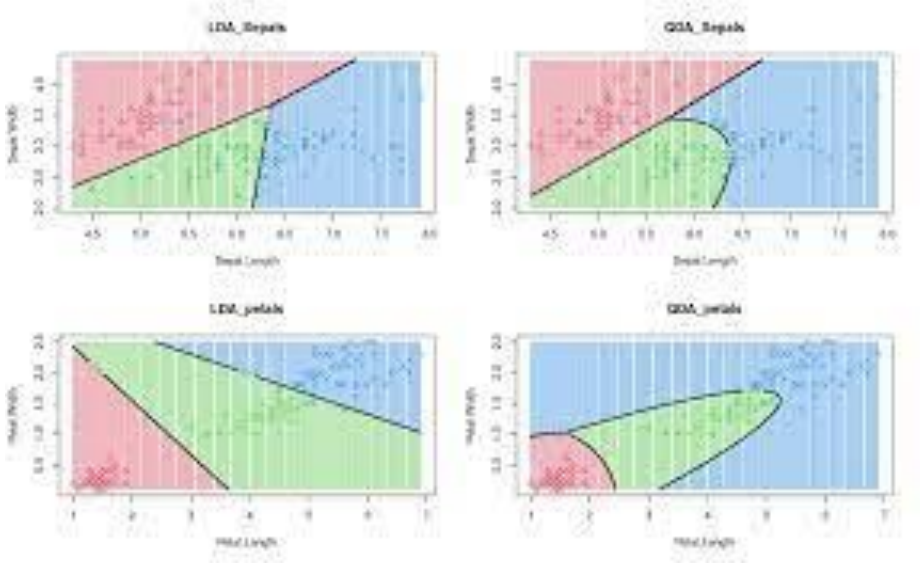


FIG 16: Quadratic discriminant analysis

HARDWARE AND SOFTWARE USED

HARDWARE REQUIREMENT

S.No.	Hardware Tools	Minimum requirements
1	Processor	I3 or above
2	Hard Disk	10GB or above
3	RAM	4GB or above
4	Monitor	17’’ coloured
5	Mouse	Optical/touchpad
6	Keyboard	122keys/laptop keyboard

SOFTWARE REQUIREMENT:

S.No.	Software Tools	Minimum Requirements
1	Operating System	Windows, Linux, Mac OS
2	Technology	Python, Machine learning,
3	Version	3.6 or above
4	Scripting Language	Python
5	IDE	Jupyter Notebook.
6	Library	Pandas, Numpy , Sklearn, Py-caret.

PREREQUISTIES FOR RUNNING THE CODE IN MACHINE

STEPS	**PROCESS**
1	Install Anaconda Navigator Python 3.8 or above
2	Install Jupyter Notebook Version 6.3.0
3	Download the dataset
4	Install Required Library

IMPLEMENTATION

Python

Python is a commonly interpreted, interactive, object-oriented, high-level programming language. Created by Guido van Rossum from 1985 to 1990. Like Perl, the Python source code is available under the GNU General Public License (GPL). Python is a high-level, interpreted, interactive, object-oriented scripting language. Python is designed to be easy to read. English keywords are often used, but other languages use punctuation and have less syntactic structure than other languages. Python is essential for students and professionals to become good software developers. Especially when working in the field of web development. List some of the main benefits of learning Python.

- Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive − You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Python is Object-Oriented − Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- Python is a Beginner's Language − Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

Characteristics of Python

Following are important characteristics of Python Programming −

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Fig 15: Python

Python Library

1.Pandas

Pandas is a software library created for the Python programming language for data manipulation and analysis. In particular, it provides data structures and operations for working with numeric tables and time series. This is free software released under the 3clause BSD license. This name comes from the term "panel data". This is an econometric term for a dataset that contains observations of the same person over multiple periods. This name means the term "Python data analysis" itself. Wes McKinney began manufacturing pandas from 2007 to 2010 as a researcher at AQR Capital

Library features

- A DataFrame object for manipulating data with a unified index.
- Storage A tool for reading and writing data between data structures and various file formats.
- Data synchronization and missing data integration process.
- Dataset reformation and pivoting.

- Label-based slices, flashy indexing, and a subset of large datasets.
- Inserts and deletes data structure columns. Grouping by engine that allows split apply combine operations on records.
  - Record merge and merge.
  - Hierarchical axis index for processing high-dimensional data in low-dimensional data structures.
  - Time series features: date range generation and frequency conversion, move window statistics, move window linear regression, date shifts and delays. Provides data filtering.
  - The library has been significantly optimized for performance, with important code paths written in Python or C.

2.Numpy

NumPy targets the CPython reference implementation of Python, a non-optimized bytecode interpreter. Mathematical algorithms written for this version of Python often run much slower than their compiled equivalents. NumPy partially addresses the slowdown problem by providing multidimensional arrays and functions and operators that process arrays efficiently. To use them, you need to rewrite your code (mainly internal loops) using NumPy.

Using NumPy in Python provides features comparable to MATLAB in that both are interpreted, and as long as most operations use arrays or matrices instead of scalars, users can write fast programs. I can do it. By comparison, MATLAB has a number of additional toolboxes (especially Simulink), but NumPy is essentially integrated with Python. Python is a more modern and complete programming language. Additional Python packages are also available. SciPy is a library that adds MATLAB-like functions, and Matplotlib is a plot package that provides MATLAB-like plot functions. Internally, both MATLAB and NumPy rely on BLAS and LAPACK for efficient calculation of linear algebra. The widely used computer vision library OpenCV's

Python binding uses NumPy arrays to store and process data. Images with multiple channels are simply represented as a 3D array, so indexing, slicing, or masking with other arrays is a very efficient way to access a particular pixel in the image. NumPy arrays as an OpenCV universal data structure for images, extracted feature points, filter cores, etc. greatly simplify programming workflows and debugging. The core function of NumPy is "ndarray" which is the data structure of n-dimensional array. These arrays are extended views of memory. In contrast to Python's built-in list data structures, these arrays are homogeneously typed. All elements of a single array must be of the same type. Such an array is also a view in the memory buffer allocated to the CPython interpreter by C / C ++, CPython, and Fortran extensions, and is constant with existing numeric libraries because it does not require copying data. It is compatible. This feature is used in the SciPy package. This package contains many such libraries (especially BLAS and LAPACK). NumPy has built-in support for memory-mapped arrays.

3.Sklearn

Scikit-learn (formerly scikits learn, also known as sklearn) is a free machine learning software library for the Python programming language. Support Vector machines, random forests, gradient boosting, kmeans, DBSCAN, and many other classification, regression, and clustering algorithms designed to work with the numerical and scientific Python libraries NumPy and SciPy. increase. Scikit-learn is a project funded by NumFOCUS.

The scikit-learn project started as scikits. learn, a Google Summer of Code project by French data scientist David Cournapeau. The name comes from the idea that it is SciPy's thirdparty extension, "SciKit" (SciPy Toolkit). The original code base was later rewritten by another developer. In 2010, Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort and Vincent Michel of the French Institute for Research in Computer Science in Rocquencourt, France, were in charge of the project and published their first publication on February 1, 2010. Of the various Scikits, Scikit-learn and Scikitimage were described in November 2012 as "well-maintained and popular." Scikit-learn is one of the most popular machine learning libraries on GitHub.

Scikit-learn is written primarily in Python and makes extensive use of NumPy for powerful linear algebra and array operations. In addition, some core algorithms are written in CPython to improve performance. Support vector machines are implemented by LIBSVM's CPython wrapper. Logistic regression and linear support vector machines via a similar wrapper for LIBLINEAR. In such cases, you may not be able to extend these methods in Python.

Scikit-learn works well with many other Python libraries such as Matplotlib, plotly for plots, NumPy for vectorization of arrays, Pandas Dataframes, SciPy and much more.

Components of scikit-learn: Scikit-learn comes loaded with a lot of features. Here are a few of them to help you understand the spread:

- Supervised learning algorithms: Think of any supervised machine learning algorithm you might have heard about and there is a very high chance that it is part of scikit-learn. Starting from Generalized linear models (e.g Linear Regression), Support Vector Machines (SVM), Decision Trees to Bayesian methods – all of them are part of scikit-learn toolbox. The spread of machine learning algorithms is one of the big reasons for the high usage of scikitlearn. I started using scikit to solve supervised learning problems and would recommend that to people new to scikit / machine learning as well.
- Cross-validation: There are various methods to check the accuracy of supervised models on unseen data using sklearn.
- Unsupervised learning algorithms: Again there is a large spread of machine learning algorithms in the offering – starting from clustering, factor analysis, principal component analysis to unsupervised neural networks.
- Various toy datasets: This came in handy while learning scikit-learn. I had learned SAS using various academic datasets (e.g. IRIS dataset, Boston House prices dataset). Having them handy while learning a new library helped a lot.
- Feature extraction: Scikit-learn for extracting features from images and text (e.g. Bag of words)

4. Py-Caret

Py-Caret is an open-supply, low-code gadget getting to know library in Python that automates gadget getting to know workflows. It is an give up-to-give up gadget getting to know and version control device that exponentially quickens the test cycle and makes you greater productive. Compared with the opposite open-supply gadget getting to know libraries, PyCaret is an exchange low-code library that may be used to update loads of traces of code with few traces only. This makes experiments exponentially rapid and efficient. PyCaret is largely a Python wrapper round numerous gadget getting to know libraries and frameworks along with scikit-learn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray, and some greater. The layout and ease of PyCaret are stimulated via way of means of the rising function of citizen statistics scientists, a time period first utilized by Gartner. Citizen Data Scientists are strength customers who can carry out each easy and reasonably state-of-the-art analytical responsibilities that might formerly have required greater technical expertise.

Py-caret is ideal for the following applications:

- Experienced data scientists who want to improve their productivity.
- Citizen data scientists who prefer low-code machine learning solutions.
- A data science expert who wants to create a rapid prototype.
- Data science and machine learning students and enthusiasts.

Features

Py-caret is Python's open source low-code machine learning library aimed at reducing hypotheses about cycle time of insights in ML experiments. This allows data scientists to perform end-to-end experiments quickly and efficiently. Compared to other open source machine learning libraries, Py-Caret is an alternative low-code library that you can use to perform complex machine learning tasks with just a few lines of code. Py-Caret is simple and easy to use.

PyCaret is an open-source machine learning library which is simple and easy to use. It helps you right from the start of data preparation to till the end of model analysis and deployment. Moreover, it is essentially a python wrapper around several machine learning libraries and frameworks such as scikit-learn, spaCy etc, It also has the support of complex machine learning algorithms which are tedious to tune and implement. So why to use Pycaret. Well, there are lots of reasons for this let me explain to you a few of them. The first Pycaret is a low-code library which makes you more productive while solving a business problem. Second Pycaret can do data preprocessing and feature engineering with a single line of code, where in reality, it is very time-consuming. Third Pycaret allows you to compare different machine learning models and finetune your model very easily. Well, there are many other advantages but for now, stick with them.

- create_app to create a basic version of the Gradio app
- create_docker for generating the requirements.txt and Dockerfile file
- create_api for making the API for regression and classification models

Time Series Module (beta)

PyCaret new time series module is now available in beta. Staying true to the simplicity of PyCaret, it is consistent with our existing API and fully loaded with functionalities. Statistical testing, model training and selection (30+ algorithms), model analysis, automated hyperparameter tuning, experiment logging, deployment on cloud, and more. All of this with only a few lines of code.

FLOWCHART

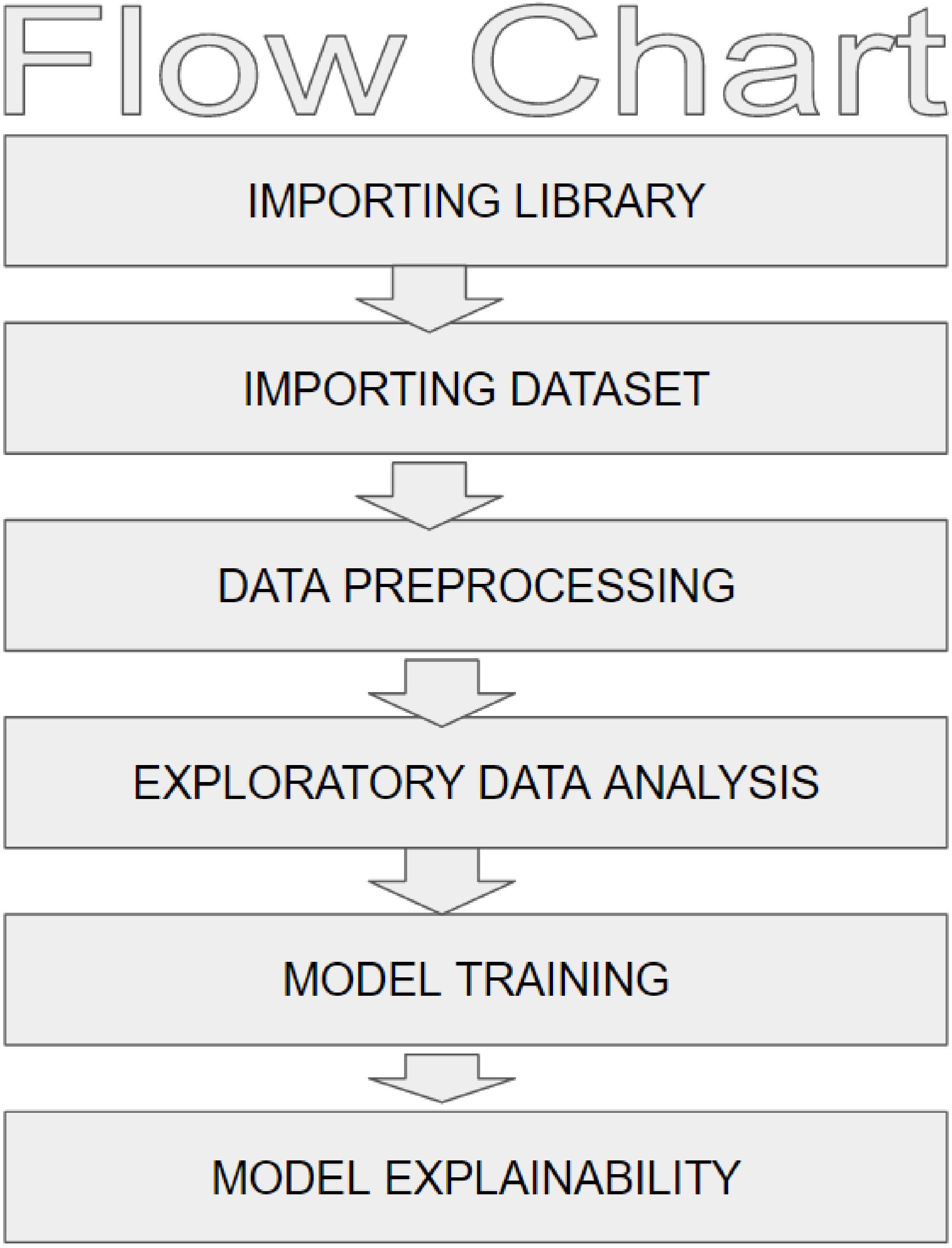


Fig 16: Depicting the flow of program

RESULTS

In this section we will observe the various graphs obtained when a plot is made between the parameters identified in the dataset. The description of the graphs obtained is prescribed in the section.

1. The following graph represents the Count plot for the distribution of the Target variable. BinaryClass is taken along x-axis and the count variable is taken along y-axis. It can be clearly inferred from the representation that patient suffering from Thyroid Disorder is much higher than the patient who are not suffering in the dataset.

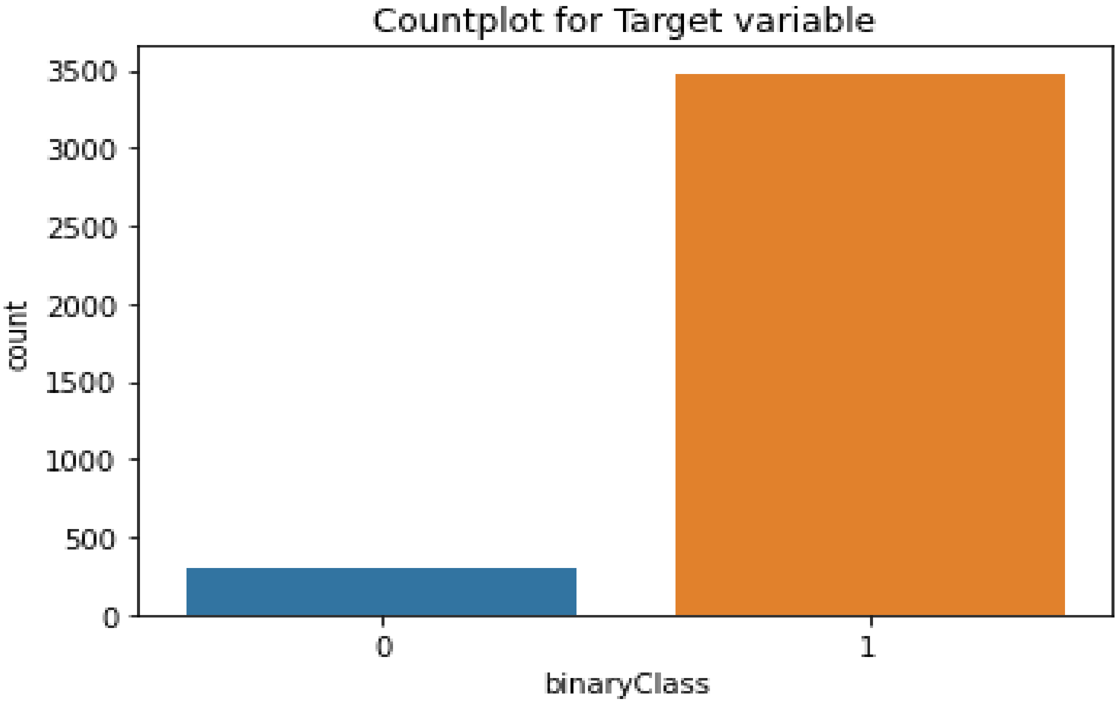


Fig 17: Countplot for Target variable

2. The following graph represents a Histogram Plot for the distribution of the Positive Class of the target variable on the basis of Age. Age is taken along x-axis and Count is taken along y-axis. It can be clearly inferred from the representation that patient having age between 40 and 60 are more prone to Thyroid Disorder.

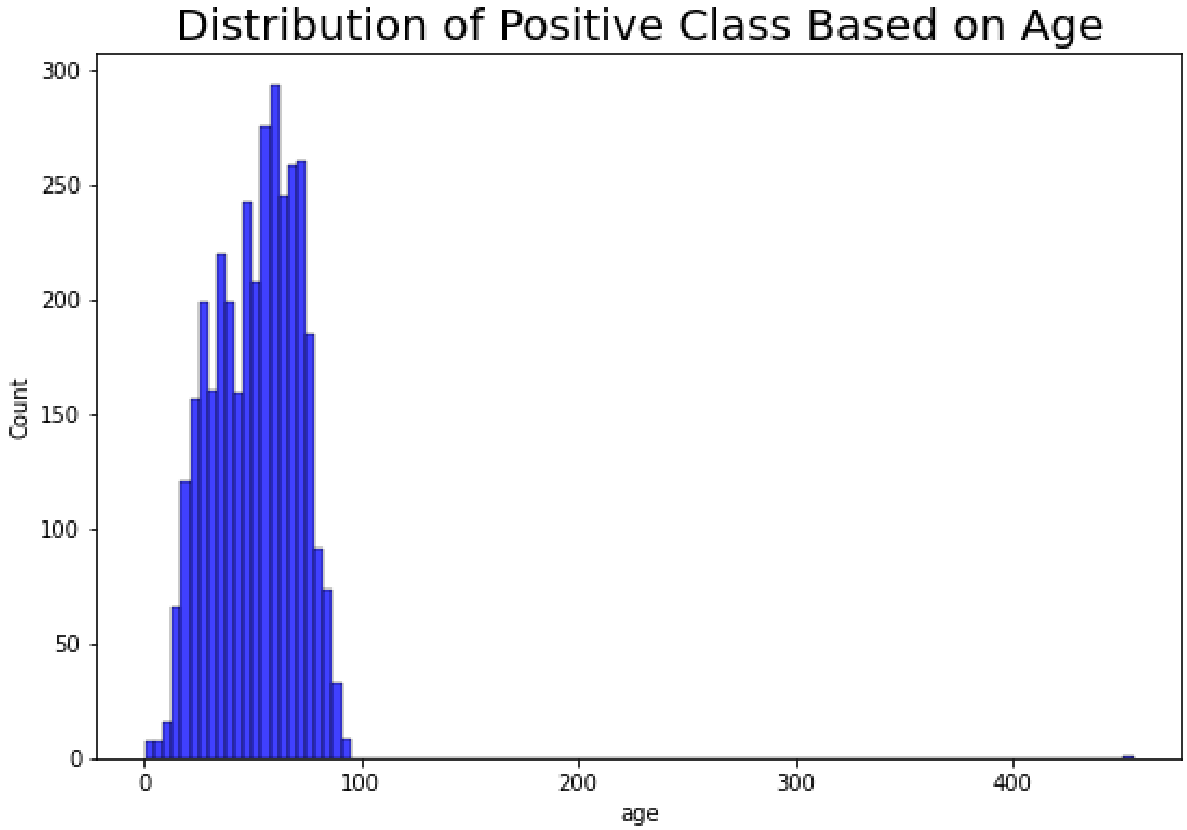


Fig 18: Distribution of Positive Class Based on Age

3. The following Pie Chart represents a Pie Graph illustrating the distribution of Sick and Well patients from the Positive class of the target variable. It can be clearly inferred from the Pie Graph that the Percentage of the Sick Patients is much higher than the patients that are well. The Sick patients comprises of the 96% of the dataset of the positive class of the target variable.

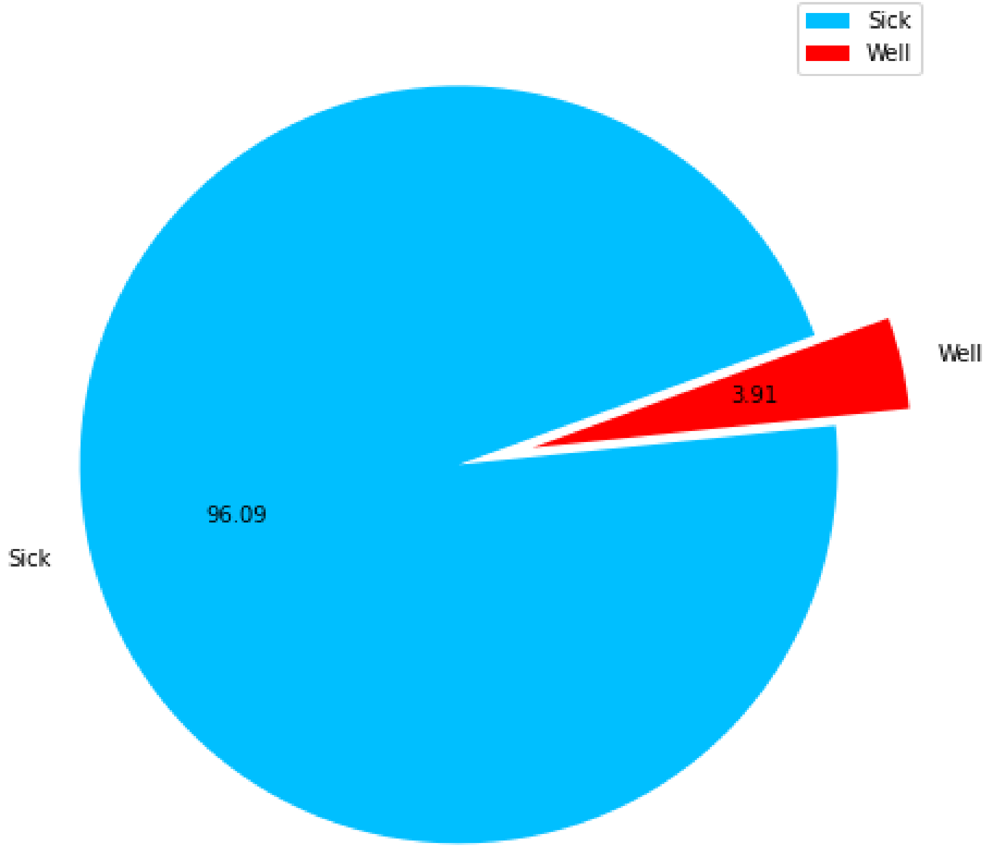


Fig 19: Distribution on the basis of sick and well

4. The following graph represents the dist plot for the distribution of the density of age of the dataset. Age is taken along x-axis and Density is taken along y-axis. It can be clearly inferred from the representation that representation that patient having age between 40 and 60 are more prone to Thyroid Disorder.

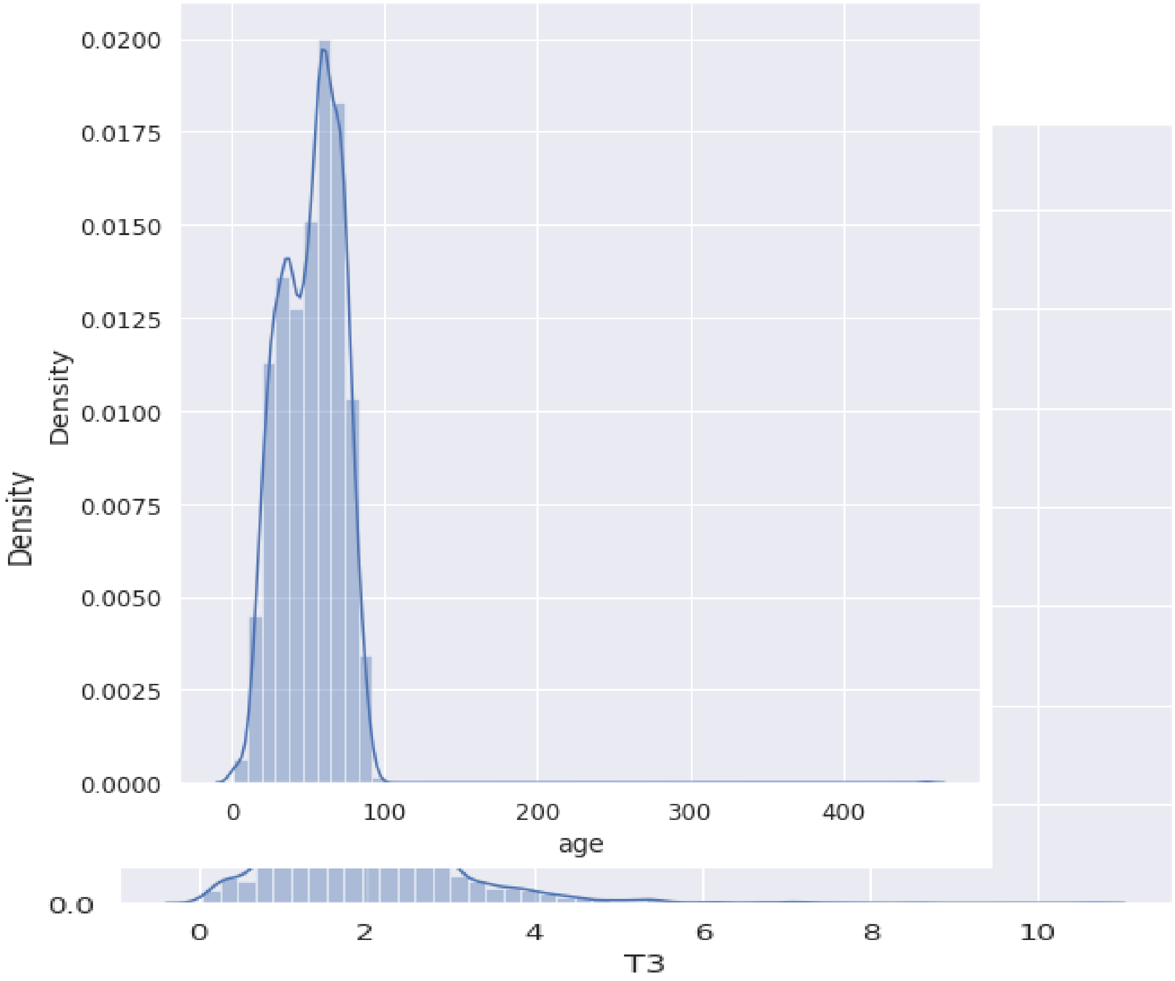


Fig 20: The dist plot on the age column

5. The following graph represents the dist plot for the distribution of the density on the basis of sex. Sex is taken along x-axis and Density is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of female patients than male patients.

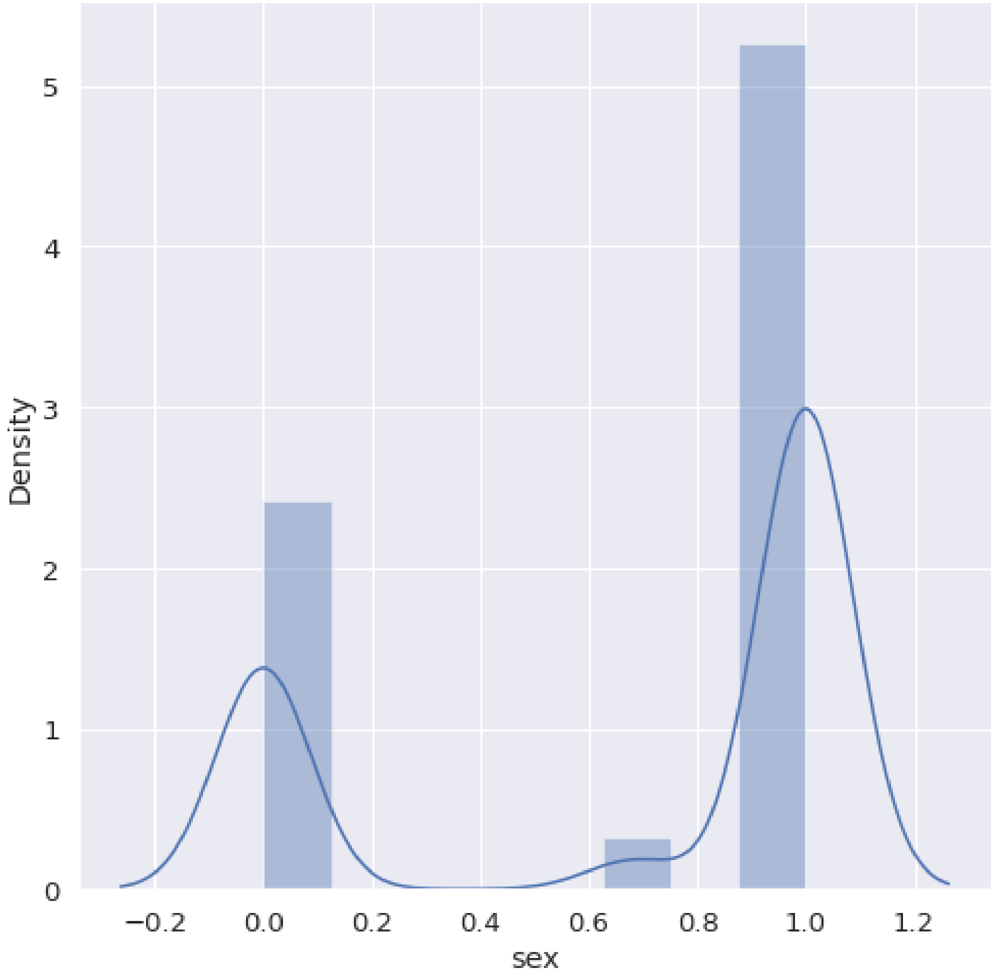


Fig 21: The dist plot on the sex column

6. The following graph represents the dist plot for the distribution of the density on the basis of T3. T3 is taken along x-axis and Density is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of patients having T3 value around 2.

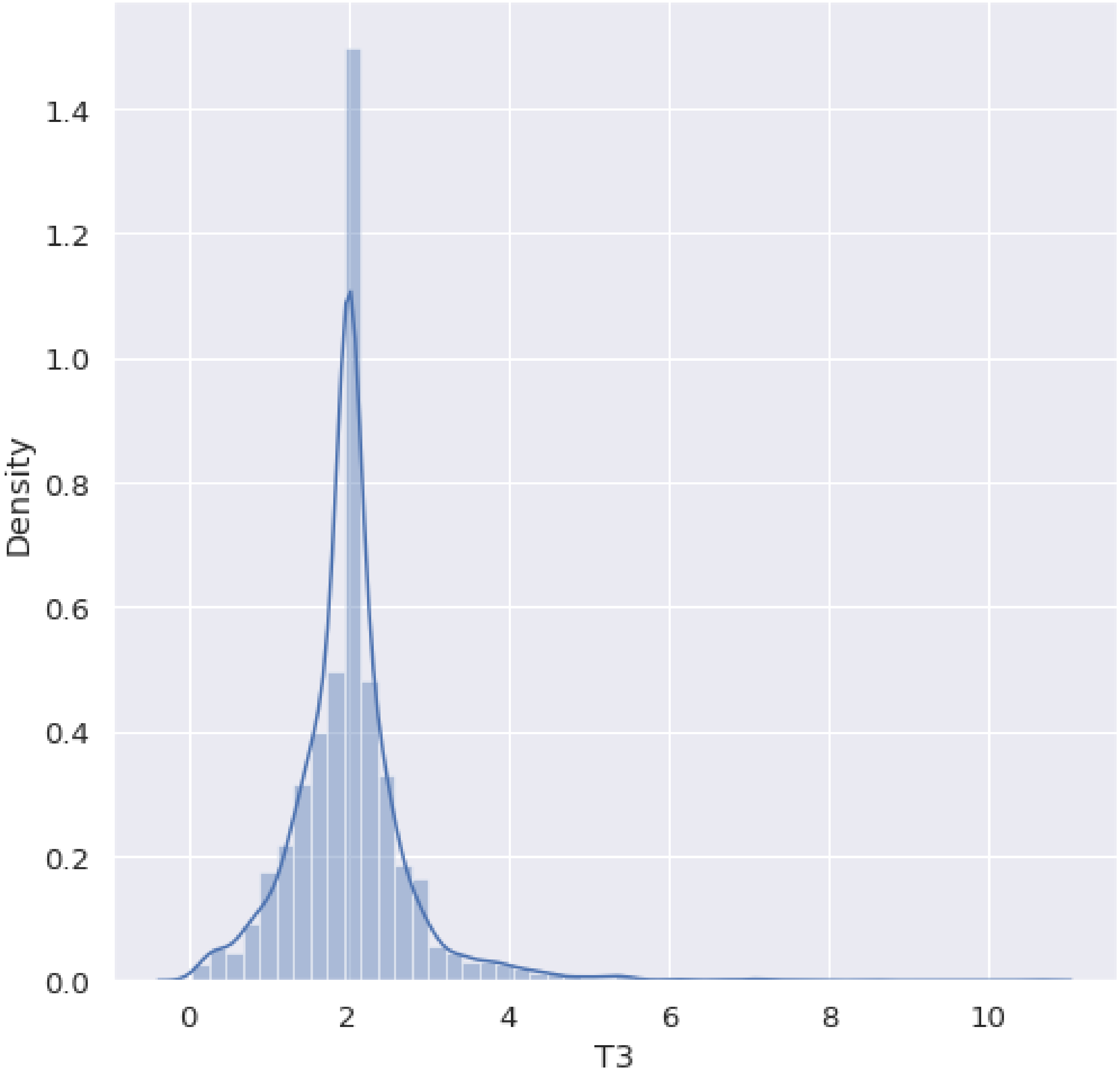


Fig 22: The dist plot on the T3 column

7. The following graph represents the dist plot for the distribution of the density on the basis of TT4. TT4 is taken along x-axis and Density is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of patients having TT4 value around 110.

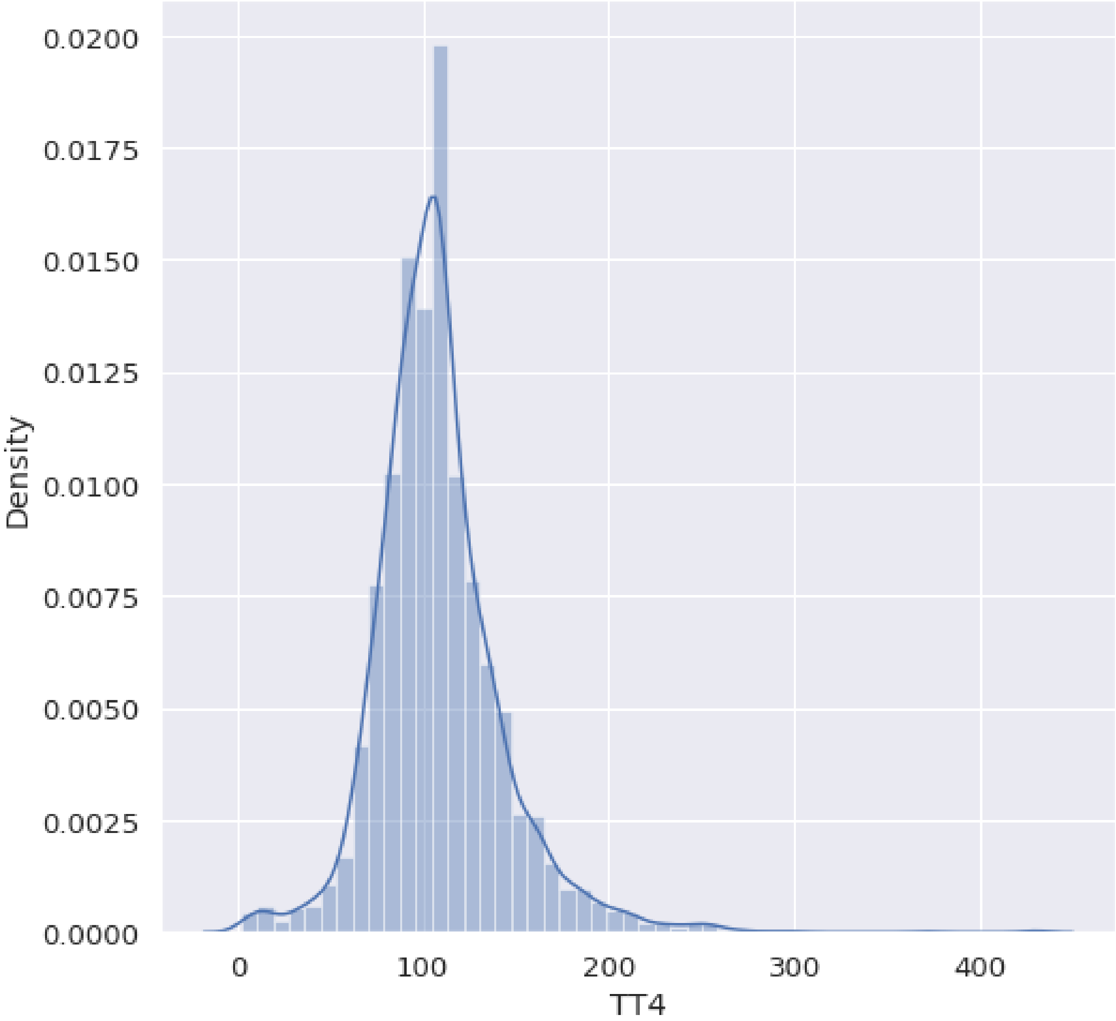


Fig 23: The dist plot on the TT4 column

8. The following graph represents the dist plot for the distribution of the density on the basis of T4U. T4U is taken along x-axis and Density is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of patients having T4U value around 1.

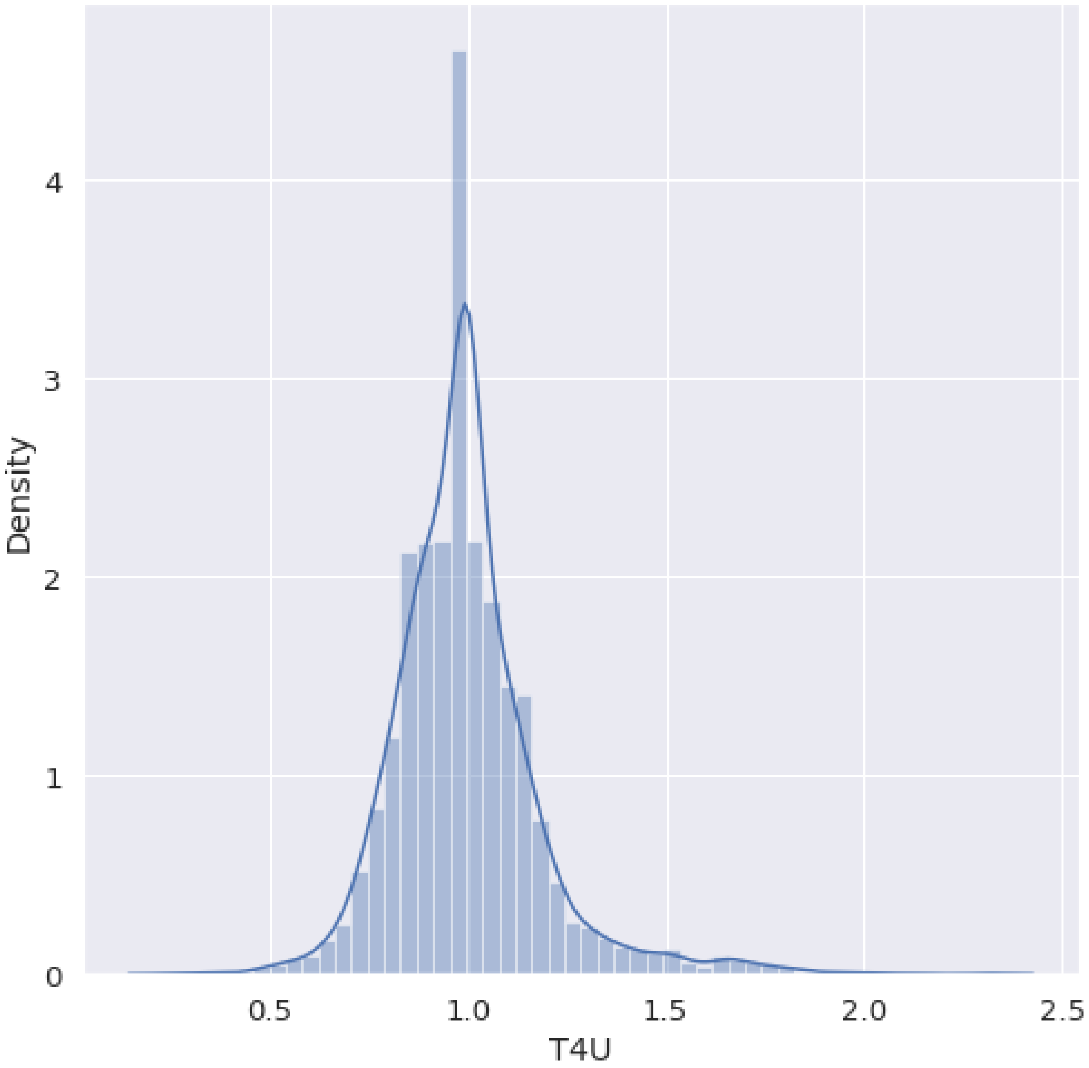


Fig 23: The dist plot on the T4U column

9. The following graph represents the dist plot for the distribution of the density on the basis of FTI. FTI is taken along x-axis and Density is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of patients having FTI value around 100.

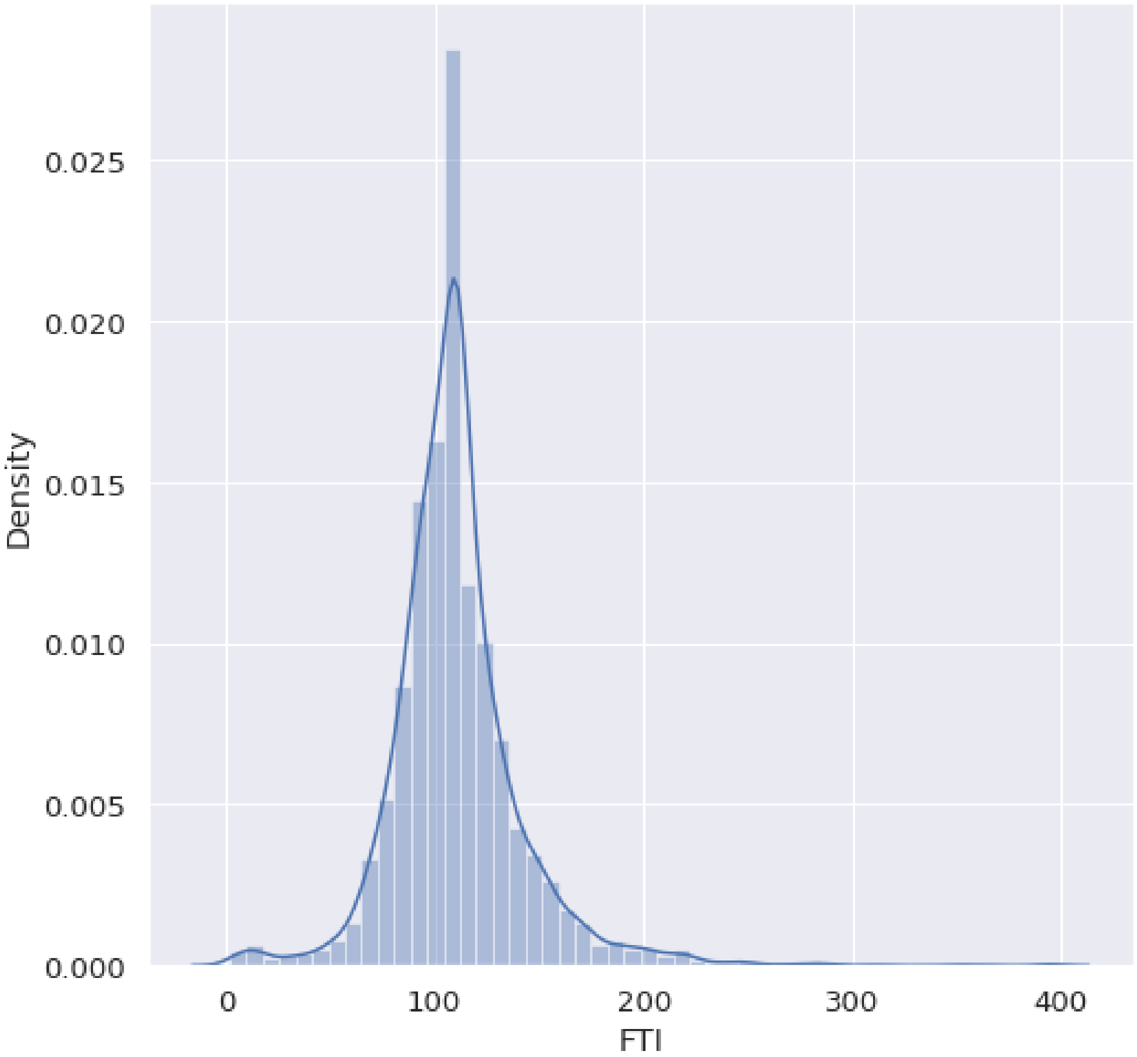


Fig 23: The dist plot on the FTI column

10. The following graph represents the dist plot for the distribution of the density on the basis of TBG measured. TBG measured is taken along x-axis and Density is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of patients having TBG measured value around 0.

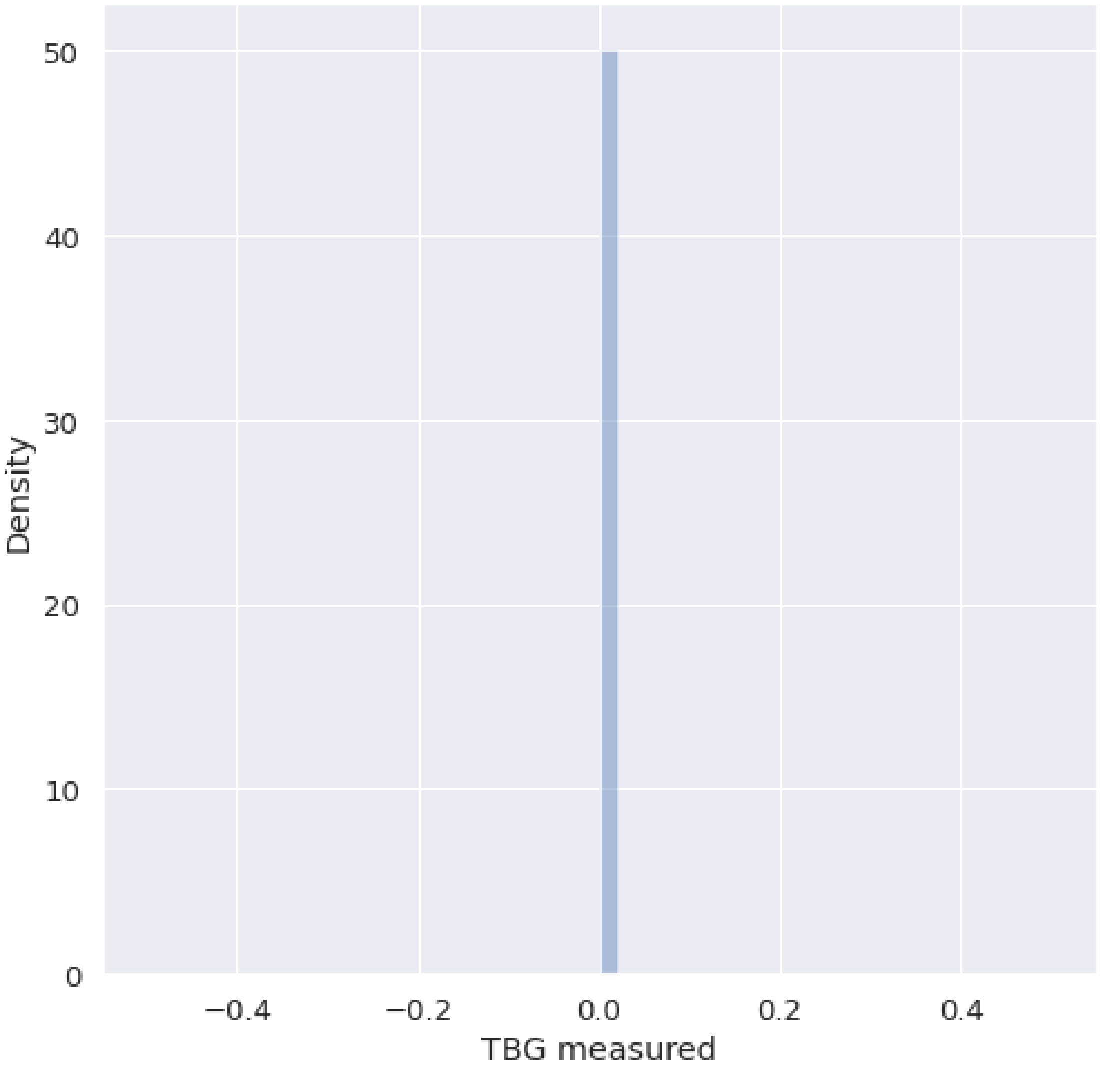


Fig 24: The dist plot on the TBG column

11. The following graph represents the joint plot (Kind =Scatter) for the distribution of the TT4 on the basis of age. Age is taken along x-axis and TT4 is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of patients having TBG measured value around 100 and 200 peaking around 120.

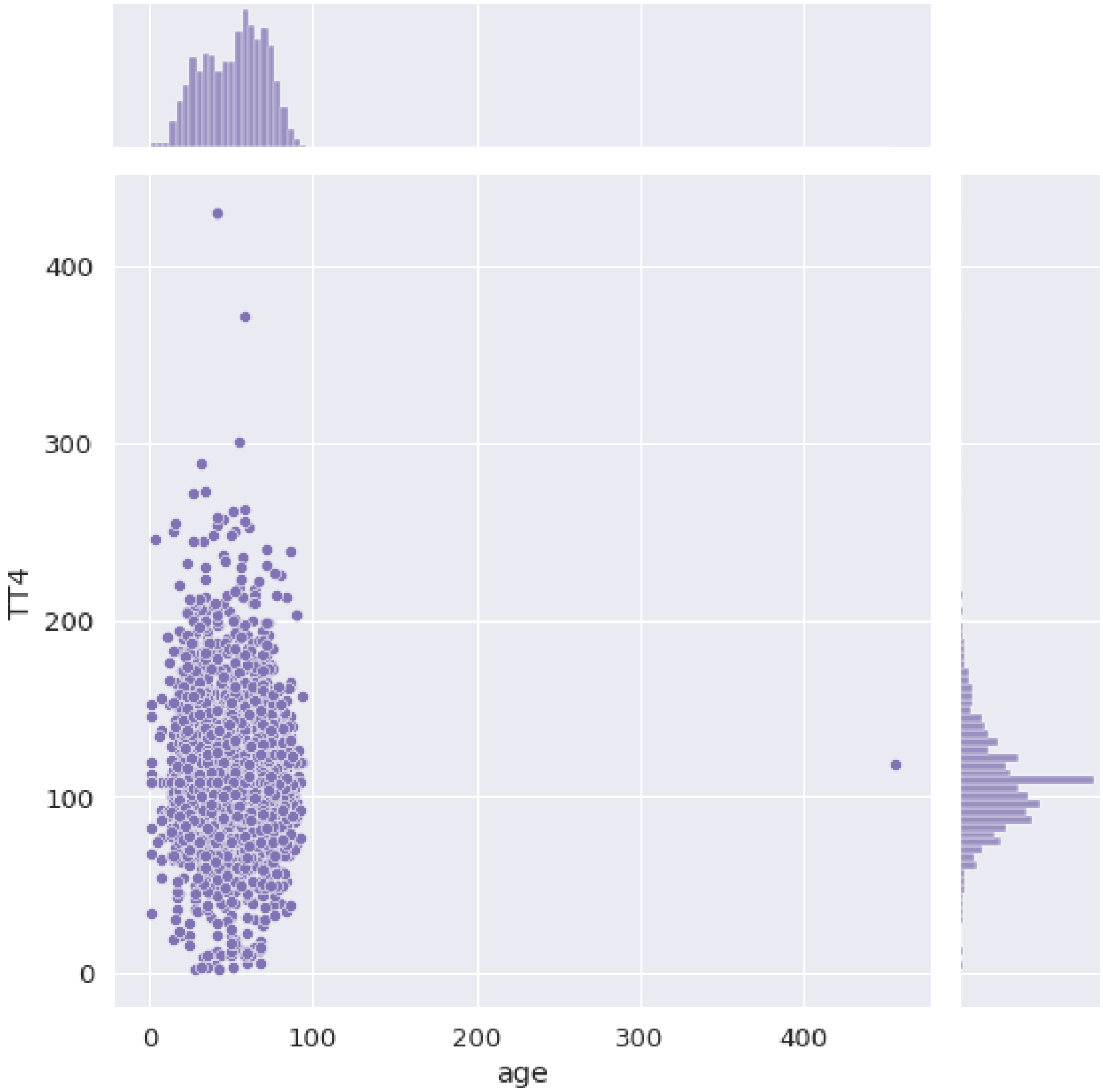


Fig 25: The joint plot on the age vs TT4

12. The following graph represents the joint plot (Kind =Scatter) for the distribution of the TT4 on the basis of age. Age is taken along x-axis and TT4 is taken along y-axis. It can be clearly inferred from the representation that the dataset contains more data of patients having TBG measured value around 100 and 200 peaking around 120.

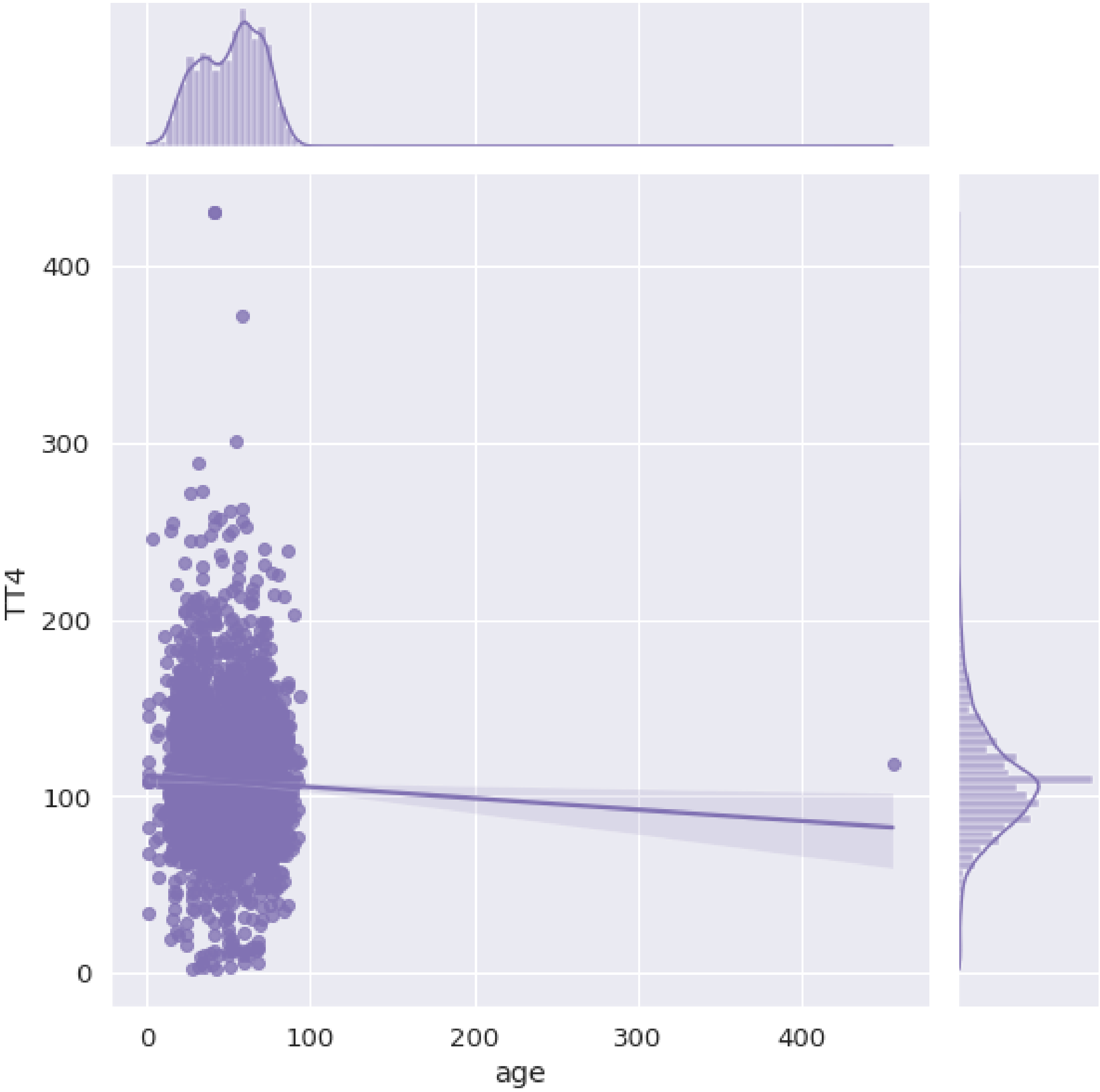


Fig 26: The joint plot on the age vs TT4

13. The following graph represents the Count plot for the distribution of the Target variable. BinaryClass is taken along x-axis and the count variable is taken along y-axis. It can be clearly inferred from the representation that patient suffering from Thyroid Disorder is much higher than the patient who are not suffering in the dataset.

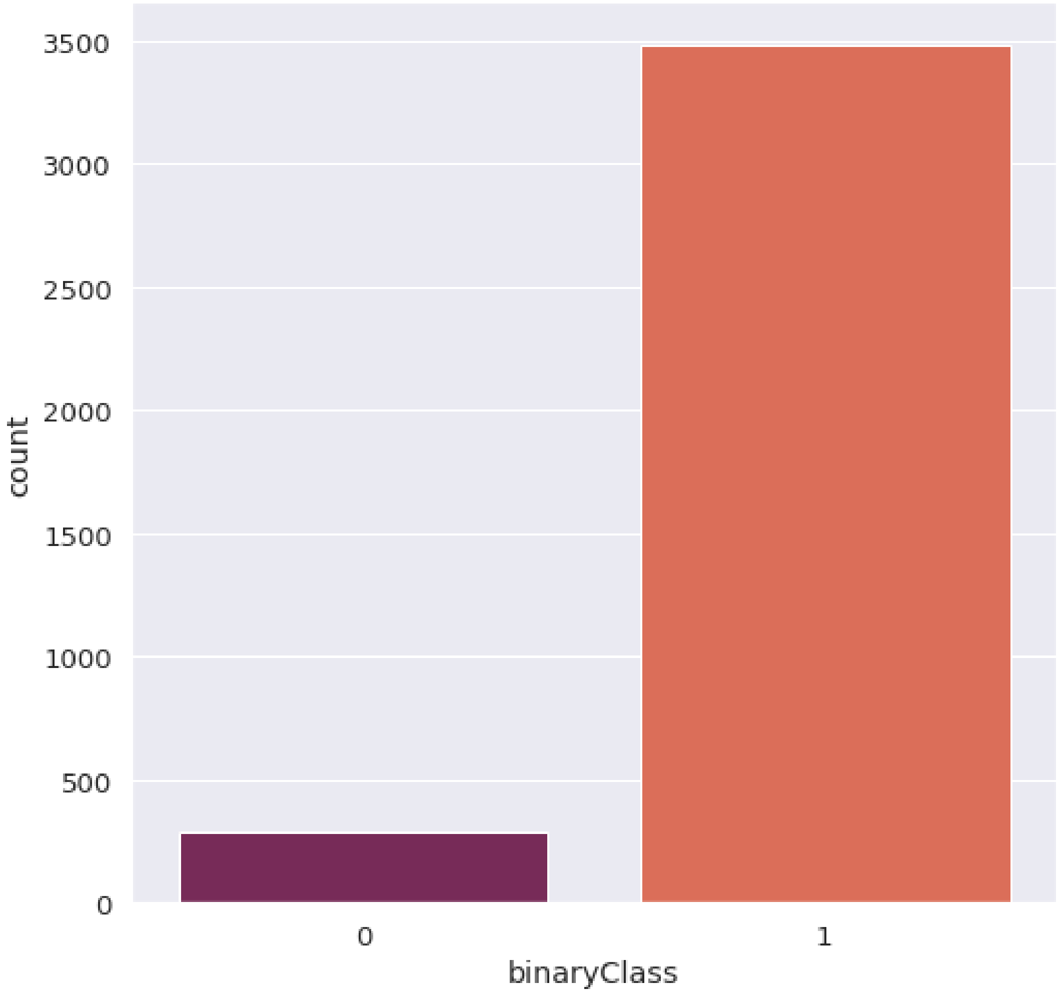


Fig 27: The count plot on the binary class

14. The following graph represents the Count plot for the distribution of the Target variable. BinaryClass is taken along x-axis and the count variable is taken along y-axis. It can be clearly inferred from the representation that female patient suffering from Thyroid Disorder is much higher than the male patient.

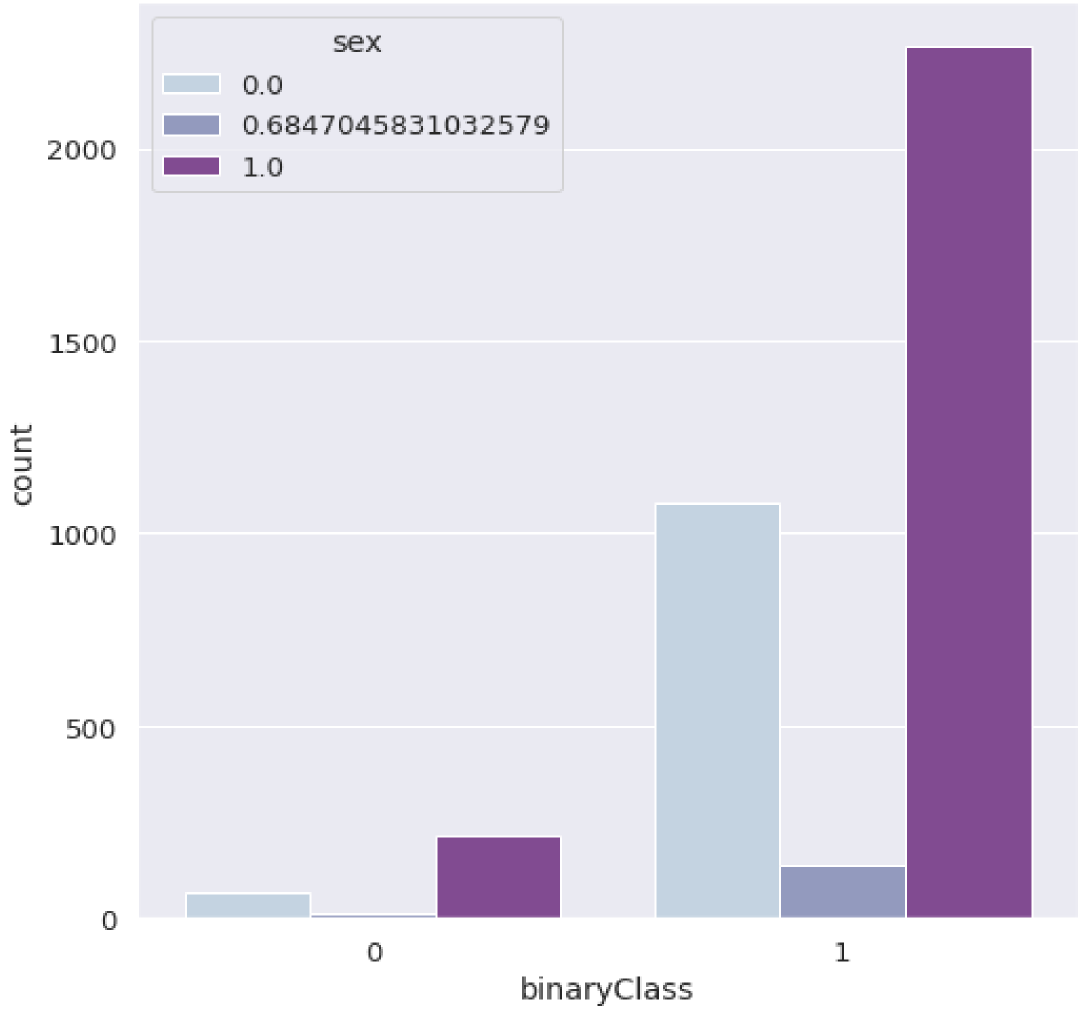


Fig 28: The count plot on binary class on the basis of sex

15. The following graph represents the Strip plot for the distribution of the Target variable. BinaryClass is taken along x-axis and the age variable is taken along y-axis. It can be clearly inferred from the representation that there is much more density of the patient suffering from thyroid disorder in the dataset.

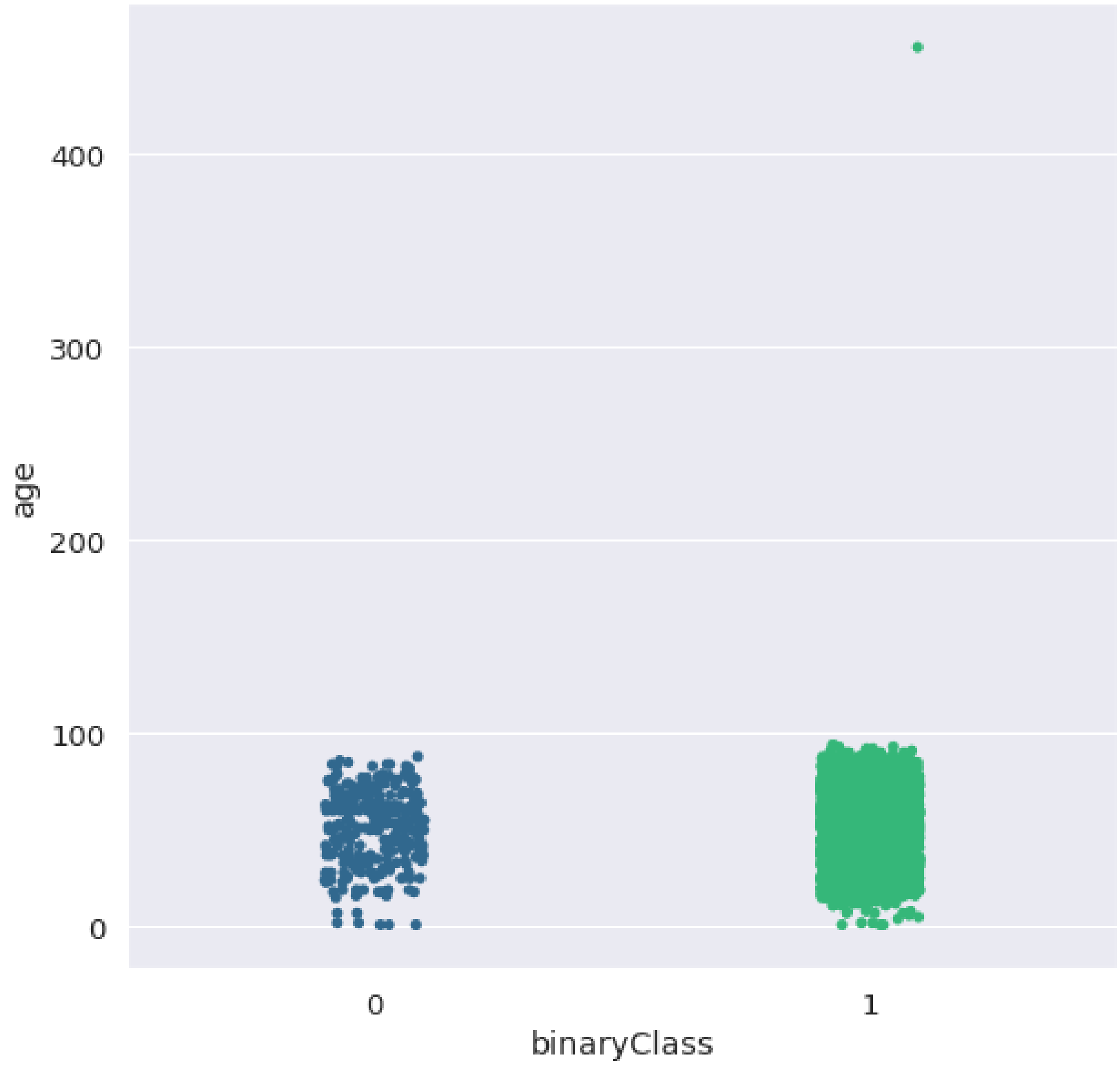


Fig 29: The strip plot on binary class

16. The following graph represents a box plot for the distribution of the age variable. BinaryClass is taken along x-axis and age is taken along y-axis. It can be clearly inferred from the representation that the age of the patient lies between 0 and 100.

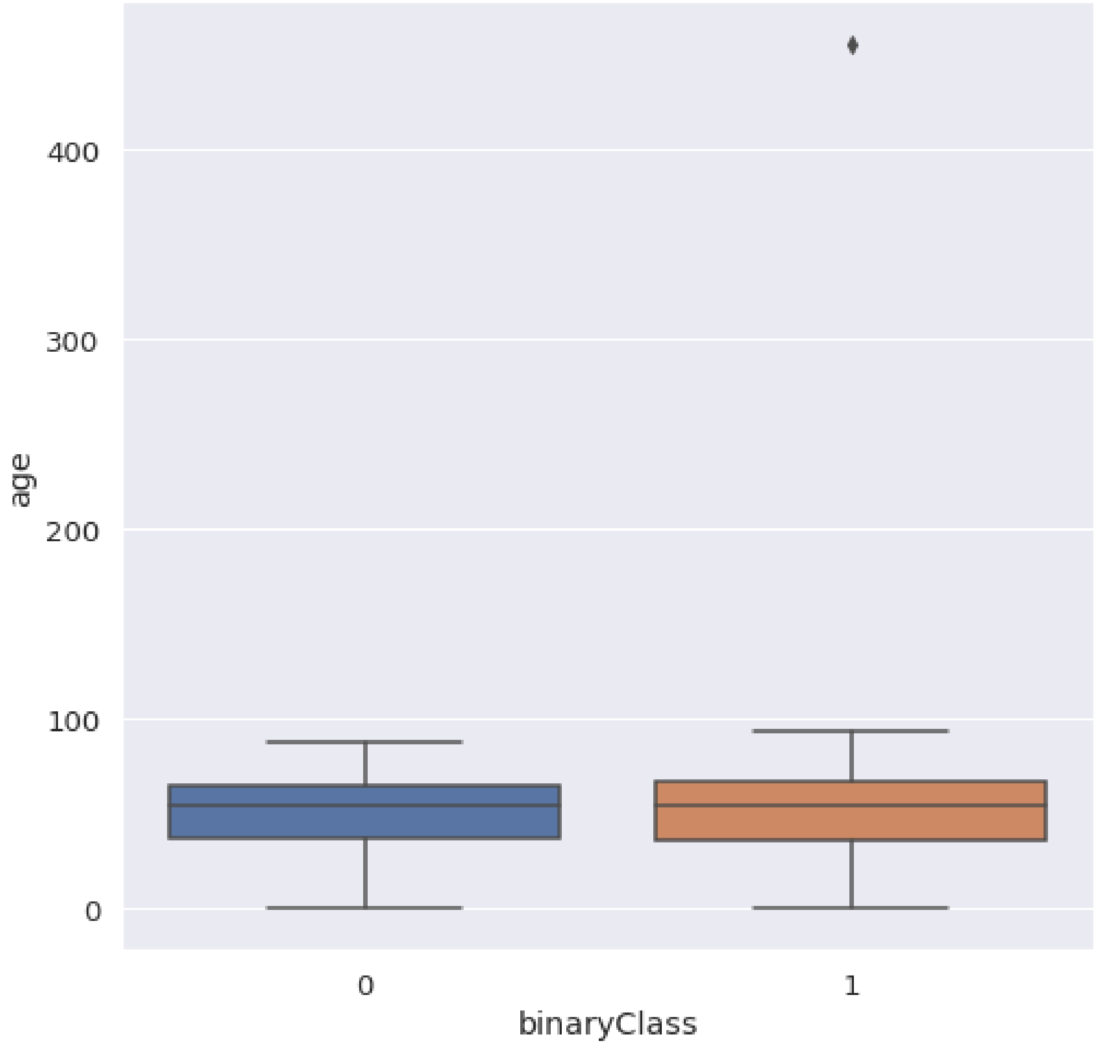


Fig 30: The box plot on binary class

17. The following graph represents the joint plot (Kind =Scatter) for the distribution of the binaryClass on the basis of FTI. FTI is taken along x-axis and binaryClass is taken along y-axis. It can be clearly inferred from the representation that the dataset contains data of FTI peaking around 120.

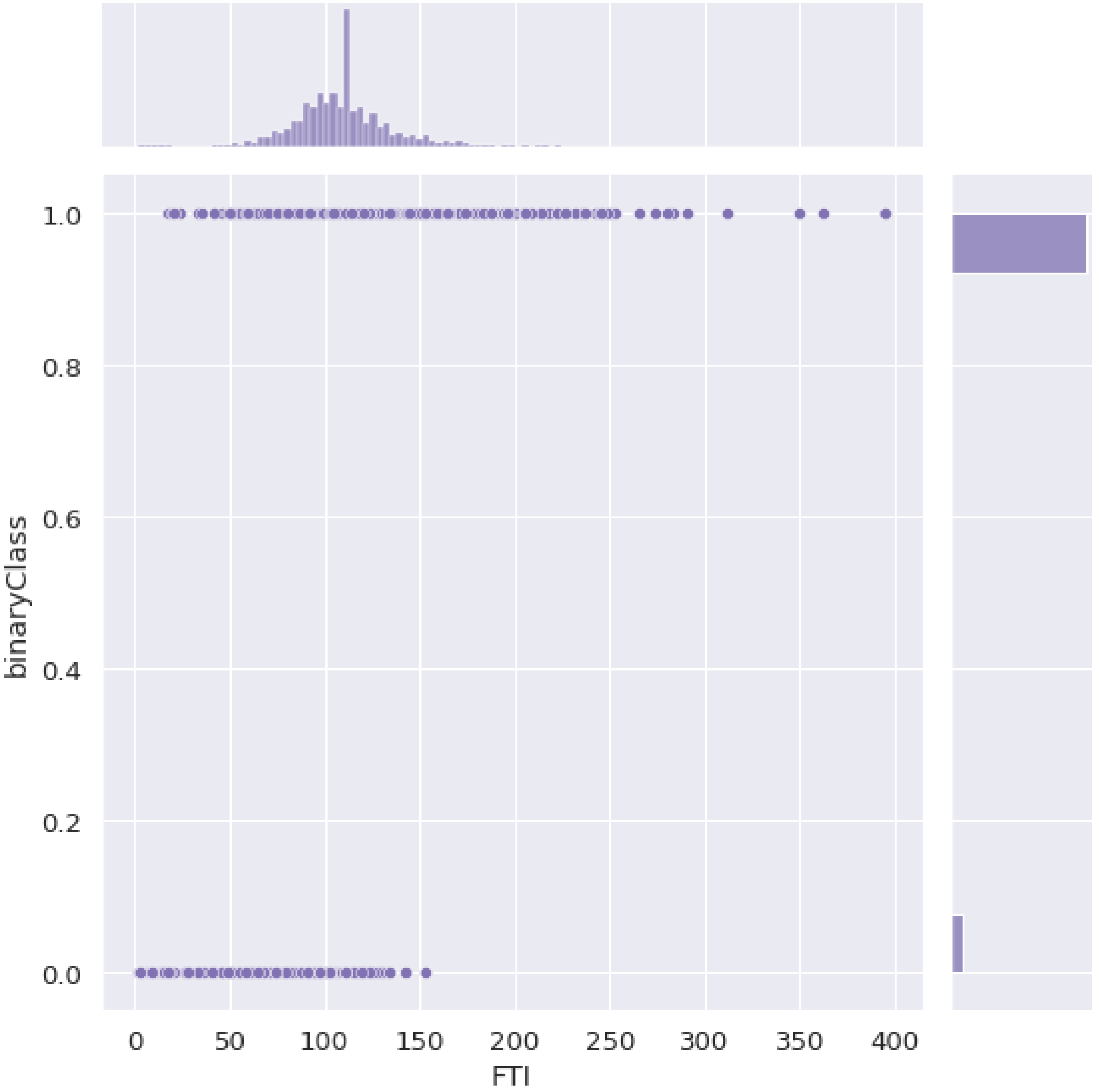


Fig 31: The joint plot on FTI vs binary class

18. The following graph represents a Heat Map representing the correlation matrix between different columns of the dataset. The Heat Map clearly states that there is high co relation between the TT4 and T4U measured.

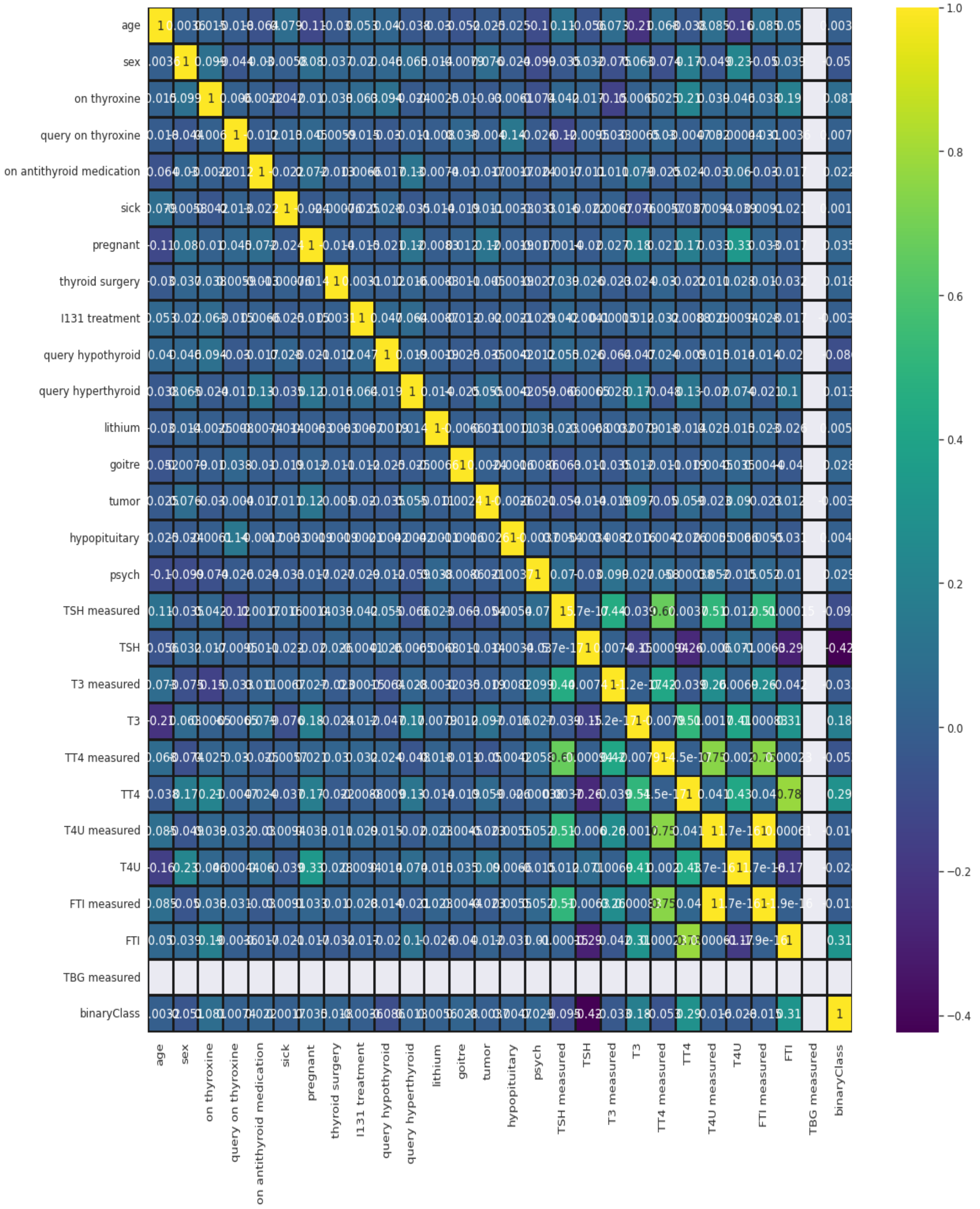


Fig 32: The heat map for co-relation

Thyroid Disorder
Algorithm

Step 1: Start

Step 2: Imported the required libraries

Step 3: Imported the DataSet.

Step 4: Preprocessing of Data

- Checking information about the dataset
- Data Cleaning

Step 5: Exploratory Data Analysis by using Seaborn and Matplotlib

- Seaborn and Matplotlib are used for plotting graphs for exploratory analysis.

Step 6: Checking Columns for Numeric Feature

- All columns having numeric features are scanned and are moved ahead in the pipeline.

Step 7: Setting Up the Columns for Model Training

- Suitable Columns are inferred and moved ahead for model training.

Step 8: Testing a Machine Learning Model for the results

Step 9: Repeat Step 8 until the best fit is found.

Step10: Best fit is chosen from Step 8

Step11: Best fit is compared again with different folds to obtain the best result.

Step12: Best fit is further hyper tuned to give the best possible result that can be obtained.

Step13: The Tuned model is finally selected and saved to the local directory.

Step14: Read the model from the local directory

Step15: Split the dataset into testing and training

Step16: Use the model to test on the testing DataSet.

Step17: Compare the Predicted and the true values.

Step18: Accuracy is Obtained.

CONCLUSION

Thyroid disorder is one of the diseases that affect the world's population, and the number of cases of this disease is increasing. Because of medical reports that show serious imbalances in thyroid diseases, our study deals with the classification of thyroid disease between hyperthyroidism and hypothyroidism. This disease was classified using algorithms. Machine learning showed us good results using several algorithms. We found the following accuracy from our study.

We worked on 14 different machine learning models. We found an accuracy of 99.55 % on the use of Gradient Boosting Classifier. We found an accuracy of 99.47 % on the use of Ada Boost Classifier. We found an accuracy of 99.47 % on the use of Light Gradient Boosting Machine. We found an accuracy of 99.39 % on the use of Decision Tree Classifier. We found an accuracy of 99.39 % on the use of Random Forest Classifier. We found an accuracy of 97.84 % on the use of Extra Trees Classifier. We found an accuracy of 96.02 % on the use of Logistic Regression. We found an accuracy of 95.61 % on the use of K Neighbours Classifier. We found an accuracy of 95.45 % on the use of SVM- Linear Kernel. We found an accuracy of 94.39 % on the use of Linear Discriminant Analysis. We found an accuracy of 93.48 % on the use of Ridge Classifier. We found an accuracy of 92.58 % on the use of Dummy Classifier. We found an accuracy of 22.35 % on the use of Naïve Bayes. We found an accuracy of 19.85 % on the use of Quadratic Discriminant Analysis.

On further tuning the Gradient Boosting Classifier, we got an accuracy of about 100 %.

This project is cost estimated in 120 days or 16 weeks and completed by four-member (three students and one professor).The total cost for this project is 10,082.

FUTURE WORK

Although We have achieved an accuracy of 100 % from our machine learning model, but the dataset that we have used do not cover the greater landscape of the problem statement. Our model can be further improved when deployed and tested on a larger dataset.

Our model works on the currently given columns in the dataset taken. If any further new column be introduced in the future, any new symptom discovered by the medical fraternity in the future, the model need to be worked again. The entire Data processing, cleaning, and visualizing process needs to be repeated again from the scratch. Although the Core logic will remain the same, but the code and model will be required to be worked upon again.

Further work of creating a graphical user interface for the following model can be done. In which if any further data is added through the GUI, that entry will get added to the dataset, and the model be developed and worked again according to the new data found.

The GUI can also contain the predicting abilities, that if anybody enters their data in the interface, then it can predict if the person is suffering from Thyroid Disorder or not.

The GUI can also contain an option for choosing from the 12 algorithms we have worked upon, and give the results according to the algorithm selected.

REFERENCES

[1] V.K. Singh, N.D. Yadav, R.K. Singh and M. Sahu, "Detection of Thyroid Using Machine Learning Approach ," NIU International Journal of Human Rights, ISSN:- 2394-0298, vol. 3, pp. 65-80, March 2022.

[2] B Gopinath and N. Shanthi, “Support Vector Machine based Diagnostic system for thyroid cancer using statistical texture features, ” Asian Pacific Journal of Cancer Prevention, Vol 14, 2013, pp.97-102.

[3] W. Gu, Y. Mao, Y. He, Z. Liang, X. Xie, Z. Zhang, and W. Fan, “High Accuracy Thyroid Tumor Image Recognition Based on Hybrid Multiple Models Optimization,” High Accuracy Thyroid Image Recognition IEEE, vol. 8, pp. 128426-128439, 2000.

[4] J. Gehrke, R. Ramakrishnan, and V. Ganti, “RainForest-A Framework for Fast Decision Tree Construction of Large Datasets,” Data Mining and Knowledge Discovery, vol. 4, pp. 127-162, 2000.

[5] Y. Zhang, M. Ni, C. Zhang, S. Liang, S. Fang, R. Li, and Z. Tan, “Research and Application of AdaBoost Algorithm Based on SVM,” IEEE 8th Joint International Technology and Artificial Conference (ITAIC 2019), pp. 662-666, 978-1-5386-8178-7, 2019.

[6] H. Luo and X. Pan, “Logistic Regression and Random Forest for Effective Imbalanced Classification,” IEEE 43rd Annual Computer Software and Application Conference (COMPSAC), pp. 916-917, 978-1-7281-2607-4, 2019.

[7] D. Wolfensberger, M. Gabella, M. Boscacci, U. Germann, and A. Berne, “RainForest: a random forest algorithm for quantitative precipitation estimation over Switzerland,” RainForest: a random forest algorithm for QPE, Vol. 14, pp. 3169-3193, 2021.

[8] X. Chai, “Diagnosis Method of Thyroid Disease Combining Knowledge Graph and Deep Learning,” Diagnosis Method of Thyroid Disease Combining Knowledge Graph and Deep Learning IEEE, Vol. 8, pp. 149787-149795, August 14, 2020.

[9] S.H.Adil, M.Ebrahim, K.Raza, S.S.A.Ali and M.A.Hashmani, “Liver Patient Classification using Logistic Regression,” 2018 4th International Conference on Computer and Information Science (ICCOINS), pp. 1-5, 978-1-5386-4744-8/18.

[10] A.Singh, S. Prakash.B and K. Chandrasekaran, “A Comparison of Linear Discriminant Analysis and Ridge Classifier on Twitter Data,” International Conference on Computing, Communication and Automation(ICCCA 2016), pp.133-138, 978-1-5090-1666-2/16

[11] H. Ding, X. K. Wang, “Research On Algorithm Of Decision Tree Induction,” Proceedings of the First International Conference on Machine Learning and Cybernetics, pp. 1062-1065, 0-7803-7508-4, November 2002.

[12] F. Abdolali, A. Shahroudnejad, A. R. Hareendranathan, J. L. Jaremko, M. Noga, and K. Punithakumar, “A Systematic review on the role of artificial intelligence in sonographic diagnosis of thyroid cancer: Past, Present and future,” IET Research Journals, pp. 1-9, 2015.

[13] H.Sifaou, A.Kammoun and M.S.Alouini, “High-Dimensional Quadratic Discriminant Analysis Under Spiked Covariance.” High-Dimensional QDA Under Spiked Covariance Model, Volume 8, June 25,2020, pp. 117313-117323, 2020.3004812.

[14] F. Alzamzami, M.Hoda, and A.E.Saddik, “Light Gradient Boosting machine for general sentiment classification on short texts: A comparative evaluation.,” LGBM for General Sentiment Classification on Short Texts,Volume 8, May 25,2020, pp. 101840-101858, 2020.2997330

[15] D.Agrawal, S.Minocha and A.K.Goel, “Gradient Boosting Based classification of ion channels,” International Conference on Computing, Communication and Intelligent Systems(ICCIS),2021, pp. 102-107,978-1-7281-8529-3/21/.

[16] Y.A.Alsariera, V.E.Adeyemo, A.O. Balogun and A.K.Alzzawi, “Ai Meta-Learners and Extra Trees algorithm for the Detection of phishing websites,” Creative Common Attribution 4.0 License ,Volume 8, August 3,2020, pp.142532-142542, 2020.3013699

[17] X. Wang, X. Wang, B. Ma, Q.Li and Y.Q. Shi, “High Precision Error prediction algorithm based on ridge regression predictor for reversible data hiding” IEEE Signal, Processing Letter, Vol. 28,2021 pp. 1125-1129, 1070-9908.

[18] J. Laaksonen and E. Oja, “Classification with learning K-Nearest neighbours.,” pp.-1480-1483, 0-7803-3210-5/96

[19] K. Mossakowski and J. Mandziuk, “Learning without human expertise: A case study of the Double Bridge Problem,” IEEE Transaction on Neural network, Vol. 20, No.2, February 2009 pp.278-299, 1045-9227.

[20] J. Ahmed and M. Soomrani, “TDTD: Thyroid Disease Type Diagnostics,” 978-1-4673-8753-8, 2018.

[21] N. Dhieb, H. Ghazzai, H. Besbes, and Y. Massoud, “Extreme Gradient Boosting Machine Learning Algorithm For Safe Auto Insurance Operations,” IEEE International Conference on Vehicular Electronics and Safety (ICVES), 978-1-7281-3473-4, 2019.

[22] F. J. Yang, “An Implementation of Naïve Bayes Classifier,” International Conference on Computational Science and Computational Intelligence (CSCI), pp. 301-306, 978-1-7281-1360-9, 2018.

[23] V.K. Singh, “Proposing Solution to XOR problem using minimum configuration MLP,” Science Direct, International Conference on Computational Modeling and Security (CMS 2016), Procedia Computer Science, 85, pp.263-270.

[24] V.K. Singh and S. Pandey, “Minimum Configuration MLP for Solving XOR problem,”Proceeding of the 10th INDIACom, IEEE Conference ID:37465, 3rd International Conference on Computing for Sustainable Global Development, 168-173, BVICAM, New Delhi, India.

[25] V.K. Singh and S. Pandey,” Proposing an Ex-NOR Solution using ANN,” Proceeding International Conference on Information, Communication and Computing Technology, JIMS, New Delhi.

[26] V.K. Singh, “Mathematical Explanation To Solution For Ex-NOR Problem Using MLFFN,” International Journal of Information Sciences and Techniques,vol. 6,pp. 105-122, 2016.

[27] V.K. Singh.,”Mathematical Analysis for Training ANNs Using Basic Learning Algorithms,” Research Journal of Computer and Information Technology Sciences, 4(7),pp. 6-13,2016.

[28] V.K. Singh and V.K. Singh, “Vector Space Model : An Information Retrieval System,” International Journal of Advanced Engineering Research and Studies, vol. 4(2), pp. 141-143.

[29] V.K. Singh and V Shah, “Minimizing Space Time Complexity in Frequent Pattern Mining by Reducing Database Scanning and Using Pattern Growth Method,”Chhattisgarh Journal of Science & Technology ISSN: 0973-7219.

[30] V.K. Singh and V.K. Singh, “The Huge Potential of Information Technology,” Proceedings of National Convention on Global Leadership: Strategies and Challenges for Indian Business, Feb pp.10-11.

[31] V.K. Singh,” Proposing pattern growth methods for frequent pattern mining on account of its comparison made with the candidate generation and test approach for a given data set,” Software Engineering, pp. 203-209, Springer, Singapore, 2019.

[32] V.K. Singh, “RSTDB & Cache Conscious Techniques for Frequent Pattern Mining,” 4th International Conference On Computer Applications In Electrical Engineering Recent Advances, CERA-09, Indian Institute of Technology, Roorkee,2010.

[33] V.K. Singh, "Designing simulators for various VLSI designs using the proposed artificial neural network model TRIVENI," IEEE, International Conference on Information, Communication, Instrumentation and Control (ICICIC), pp.1-6, 2017.

[34] V.K. Singh, "Analysis of Stability and Convergence on Perceptron Convergence Algorithm," pp.149-161, International Conference by JIMS Delhi.

[35] V.K. Singh, A. Baghel, N.D. Yadav, M. Sahu and M. Jaiswal, "Machine Learning approach to detect Breast Cancer," Design Engineering (Toronto), Volume 2021, Issue-08, pp. 7054-7060, ISSN: 0011-9342, 2021.

[36] V.K. Singh, "SVM using rbf as kernel for Diagnosis of Breast Cancer," International Conference on Innovative Research in Science, Management and Technology (ICIRSMT 2021), Department of Computer Science and Application, Atal Bihari Vajpayee University, Bilaspur (C.G.), India in association with American Institute of Management and Technology (AIMT), USA, December 27-28 2021.

[37] V.K. Singh, "Support Vector Machine using rbf, polynomial, linear and sigmoid as kernel to detect Diabetes Cases and to make a Comparative Analysis of the Models," International Conference on Innovative Research in Science, Management and Technology (ICIRSMT 2021), Department of Computer Science and Application, Atal Bihari Vajpayee University, Bilaspur (C.G.), India in association with American Institute of Management and Technology (AIMT), USA, December 27-28 2021.

[38] V.K. Singh, "Colorization of old gray scale images and videos using deep learning," Published in The Journal of Oriental Research Madras, ISSN: 0022-3301, 2021.

[39] V.K. Singh, "Dual Secured Data Transmission using Armstrong Number and Color Coding," Prestige e-Journal of Management and Research, Volume 3, Issue 1, ISSN: 2350-1316, April 2016.

[40] V.K. Singh, A. Baghel and S.K. Negi, "Finding New Framework for Resolving Problems in Various Dimensions by the use of ES : An Efficient and Effective Computer Oriented Artificial Intelligence Approach," Volume 4, No. 11, ISSN(Paper): 2222-1727, ISSN(Online): 2222-2871, 2013.

[41] Chandrashekhar, R. Chauhan and V.K. Singh," Twitter Sentiment Analysis," ISPEC 8TH INTERNATIONAL CONFERENCE ON AGRICULTURAL, ANIMAL SCIENCE AND RURAL DEVELOPMENT, BINGOL, TURKEY, DECEMBER 24-25, 2021.

[42] P. Kumari, R. Gupta, S. Kumar and V.K. Singh," ML Approach for Detection of Lung Cancer," ISPEC 8TH INTERNATIONAL CONFERENCE ON AGRICULTURAL, ANIMAL SCIENCE AND RURAL DEVELOPMENT, BINGOL, TURKEY, DECEMBER 24-25, 2021.

[43] P. Sailokesh, S. Jupudi, I.K. Vamsi and V.K. Singh," Automatic Number Plate Recognition," ISPEC 8TH INTERNATIONAL CONFERENCE ON AGRICULTURAL, ANIMAL SCIENCE AND RURAL DEVELOPMENT, BINGOL, TURKEY, DECEMBER 24-25, 2021.

[44] Y.K. Reddy, K.M. Yadav and V.K. Singh," Human Activity Recognition," ISPEC 8TH INTERNATIONAL CONFERENCE ON AGRICULTURAL, ANIMAL SCIENCE AND RURAL DEVELOPMENT, BINGOL, TURKEY, DECEMBER 24-25, 2021.

[45] R.N.R.K. Prasad, P.S.S.R Ram, S. Dinesh and V.K. Singh," Text Summarization," ISPEC 8TH INTERNATIONAL CONFERENCE ON AGRICULTURAL, ANIMAL SCIENCE AND RURAL DEVELOPMENT, BINGOL, TURKEY, DECEMBER 24-25, 2021.

[46] V.K. Singh, N.D. Yadav and R.K. Singh," Diagnosis of Breast Cancer using SVM taking polynomial as Kernel," Design Engineering (Toronto), Volume 2021, Issue-08, pp. 6589-6599, ISSN: 0011-9342, 2021.

[47]https://www.google.com/search?q=decision+tree+classification&sxsrf=APq-WBsgz6myaJstSj0otH0HCrbwgJpSw:1649142000801&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjX45DZrPz2AhWrILcAHeSaCdUQ_AUoAXoECAEQAw&biw=1536&bih=722&dpr=1.25#imgrc=58fT1nQdPsx_wM

[48]https://www.google.com/search?q=2.+Random+Forest+Classifier-&sxsrf=APq-WBtekegzHyVUrEyOO2oIDaJ8UwKgdA:1649142184208&source=lnms&tbm=isch&sa=X&ved=2ahUKEwifpcuwrfz2AhW4ILcAHRjtBa8Q_AUoAnoECAEQBA&biw=1536&bih=722&dpr=1.25#imgrc=i-PpUHcg5ybztM

[49]https://www.google.com/search?q=3.+Extra+trees+Classifier-&sxsrf=APq-WBv568Ak-XOHXxPgLXiQGoMx5lAEAw:1649142697409&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj7vqalr_z2AhUE73MBHStBC0YQ_AUoA3oECAEQBQ&biw=1536&bih=722&dpr=1.25#imgrc=_QgJD6WssVTh_M

[50]https://www.google.com/search?q=4.+Logistic+Regression-&sxsrf=APqWBu5uKImYA6UbP6p9Oq4SiQFSM8Sg:1649142830863&source=lnms&tbm=isch&sa=X&ved=2ahUKEwj49_fkr_z2AhVk4HMBHYiAAOMQ_AUoAXoECAEQAw&cshid=1649142926966925&biw=1536&bih=722&dpr=1.25#imgrc=XtCYxr72xFu9uM

[51] https://www.google.com/search?q=k-neighbor+classification&sxsrf=APq-WBtAmix5aYrdp7L7NCCMgk_I1aR91Q:1649143472240&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiYueKWsvz2AhXJIbcAHXnrBBgQ_AUoA3oECAIQBQ&biw=1536&bih=722&dpr=1.25#imgrc=_wWUhdyKGVGvWM

[52]https://www.google.com/search?q=svm+linear+kernel&sxsrf=APqWBuR31R8g43OJYlrRWXUIxORjBfwaw:1649143624977&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiy0szfsvz2AhWTjeYKHSlPCycQ_AUoAXoECAEQAw&biw=1536&bih=722&dpr=1.25#imgrc=DZwD9zcslQ-tSM

[53]https://www.google.com/search?q=linear+discriminant+analysis&sxsrf=APq-WBt-qKRqso0wY2xcp1OniC73PEIfTg:1649144033393&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjLyqyitPz2AhXFLcAHYeeB8IQ_AUoAnoECAIQBA&biw=1536&bih=722&dpr=1.25#imgrc=kp9DSevj6EfTcM

[54]https://www.google.com/search?q=ridge+regression&sxsrf=APqWBuvzeAvVxYkE2Y2_ObMpbd5ZP5FJw:1649144133719&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjH7pfStPz2AhWd63MBHbGjDzEQ_AUoAXoECAIQAw&biw=1536&bih=722&dpr=1.25#imgrc=tvCzplnTEbBMNM

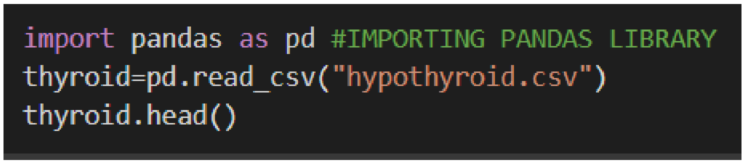
[55]https://www.google.com/search?q=dummy+classifier&sxsrf=APqWBtoN5n_Iqy5ASkXtviJmzggbsX62A:1649144210121&source=lnms&tbm=isch&sa=X&sqi=2&ved=2ahUKEwiY1872tPz2AhVJK80KHc33AZQQ_AUoA3oECAEQBQ&biw=1536&bih=722&dpr=1.25#imgrc=mYWTV_PKnitmMM

[56]https://www.google.com/search?q=naive+bayes&sxsrf=APq-WBuj_4Ar5NkDN-SMGIQthFU9JlXwGA:1649144302151&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiVnsCitfz2AhV0muYKHevBBIoQ_AUoAXoECAEQAw&biw=1536&bih=722&dpr=1.25#imgrc=pb1BfVA8XDDpzM

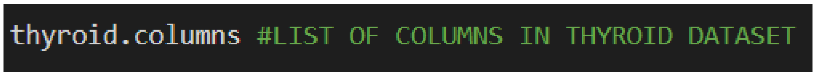
[57]https://www.google.com/search?q=Gradient+Boosting+Classifier-&sxsrf=APq-WBuKtOkiyRidki38AMlm1-V8t2blHg:1649144357166&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiUi968tfz2AhV1IbcAHWvcDiAQ_AUoAXoECAIQAw&biw=1536&bih=722&dpr=1.25#imgrc=auBXn7rZ10FR5M

[58]https://www.google.com/search?q=ada+boosting+classifier&sxsrf=APqWBvsPdnK_YBGmu54_0KYdJSbWsyeQ:1649144453311&source=lnms&tbm=isch&sa=X&ved=2ahUKEwjmo8rqtfz2AhUSheYKHTTUAW4Q_AUoAXoECAEQAw&biw=1536&bih=722&dpr=1.25#imgrc=GyTKtrbGXMM6lM

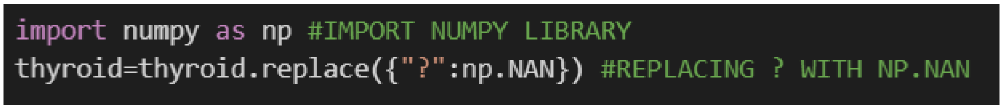
CODE



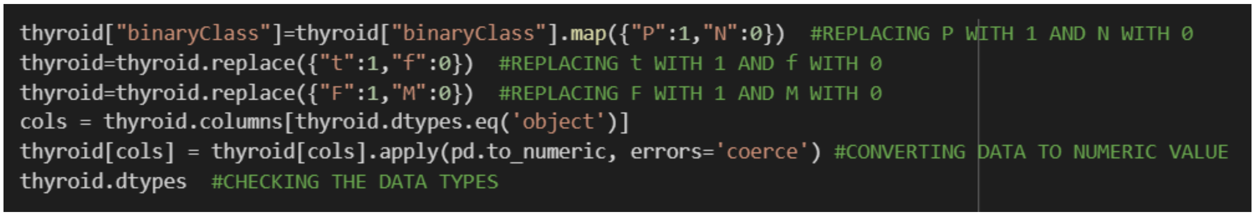
```
import pandas as pd #IMPORTING PANDAS LIBRARY
thyroid=pd.read_csv("hypothyroid.csv")
thyroid.head()
```



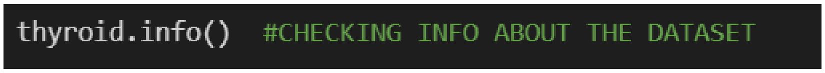
```
thyroid.columns #LIST OF COLUMNS IN THYROID DATASET
```



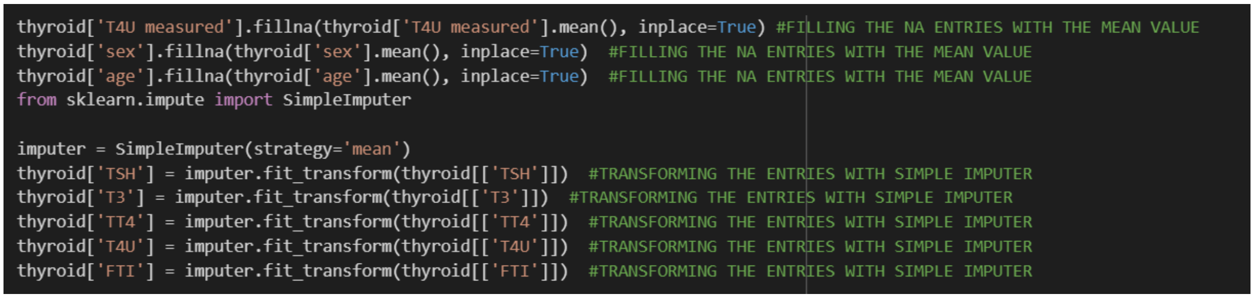
```
import numpy as np #IMPORT NUMPY LIBRARY
thyroid=thyroid.replace({"?":np.NAN}) #REPLACING ? WITH NP.NAN
```



```
thyroid["binaryClass"]=thyroid["binaryClass"].map({"P":1,"N":0})  #REPLACING P WITH 1 AND N WITH 0
thyroid=thyroid.replace({"t":1,"f":0})  #REPLACING t WITH 1 AND f WITH 0
thyroid=thyroid.replace({"F":1,"M":0})  #REPLACING F WITH 1 AND M WITH 0
cols = thyroid.columns[thyroid.dtypes.eq('object')]
thyroid[cols] = thyroid[cols].apply(pd.to_numeric, errors='coerce') #CONVERTING DATA TO NUMERIC VALUE
thyroid.dtypes  #CHECKING THE DATA TYPES
```

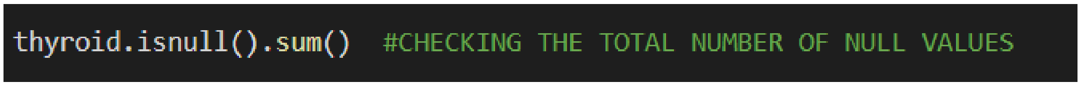


```
thyroid.info()  #CHECKING INFO ABOUT THE DATASET
```

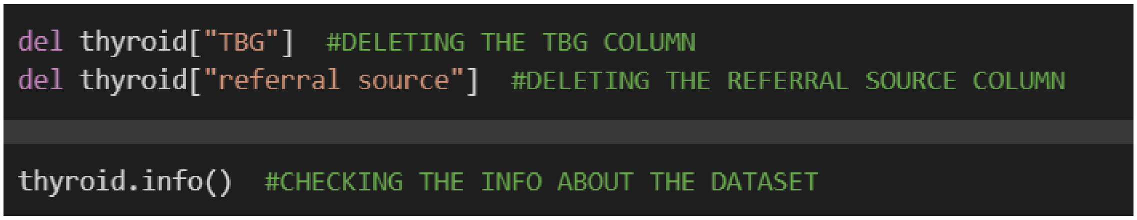


```
thyroid['T4U measured'].fillna(thyroid['T4U measured'].mean(), inplace=True) #FILLING THE NA ENTRIES WITH THE MEAN VALUE
thyroid['sex'].fillna(thyroid['sex'].mean(), inplace=True)  #FILLING THE NA ENTRIES WITH THE MEAN VALUE
thyroid['age'].fillna(thyroid['age'].mean(), inplace=True)  #FILLING THE NA ENTRIES WITH THE MEAN VALUE
from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='mean')
thyroid['TSH'] = imputer.fit_transform(thyroid[['TSH']])  #TRANSFORMING THE ENTRIES WITH SIMPLE IMPUTER
thyroid['T3'] = imputer.fit_transform(thyroid[['T3']])  #TRANSFORMING THE ENTRIES WITH SIMPLE IMPUTER
thyroid['TT4'] = imputer.fit_transform(thyroid[['TT4']])  #TRANSFORMING THE ENTRIES WITH SIMPLE IMPUTER
thyroid['T4U'] = imputer.fit_transform(thyroid[['T4U']])  #TRANSFORMING THE ENTRIES WITH SIMPLE IMPUTER
thyroid['FTI'] = imputer.fit_transform(thyroid[['FTI']])  #TRANSFORMING THE ENTRIES WITH SIMPLE IMPUTER
```

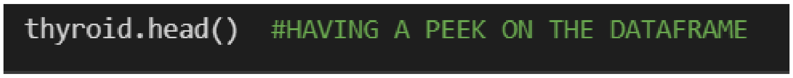


```
thyroid.isnull().sum()  #CHECKING THE TOTAL NUMBER OF NULL VALUES
```

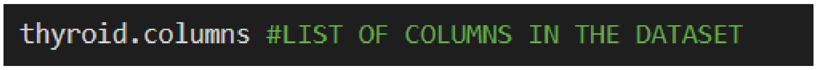


```
del thyroid["TBG"]  #DELETING THE TBG COLUMN
del thyroid["referral source"]  #DELETING THE REFERRAL SOURCE COLUMN
```

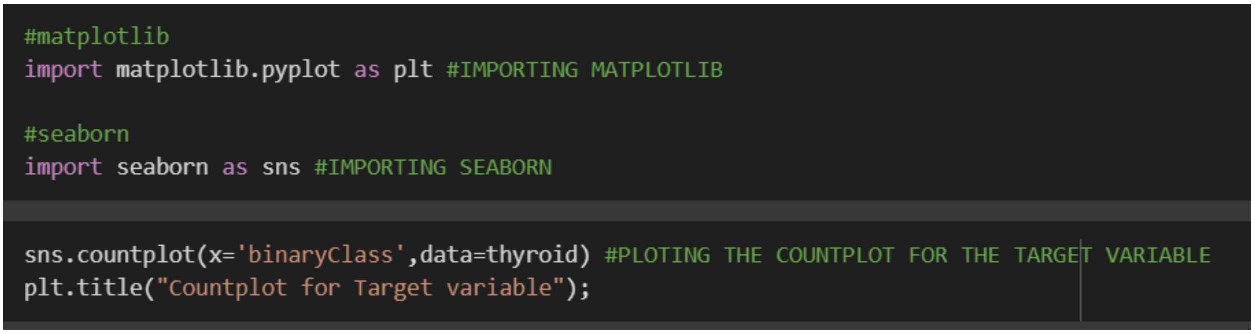
```
thyroid.info()  #CHECKING THE INFO ABOUT THE DATASET
```



```
thyroid.head()  #HAVING A PEEK ON THE DATAFRAME
```



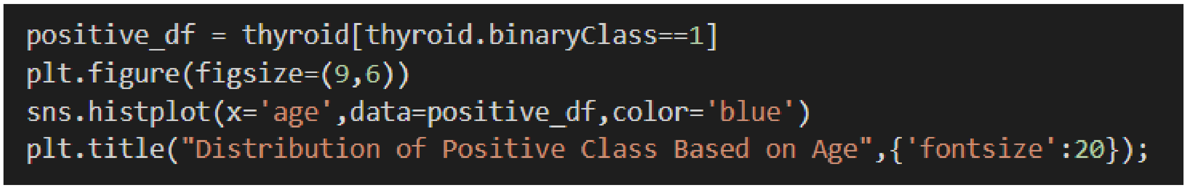
```
thyroid.columns #LIST OF COLUMNS IN THE DATASET
```



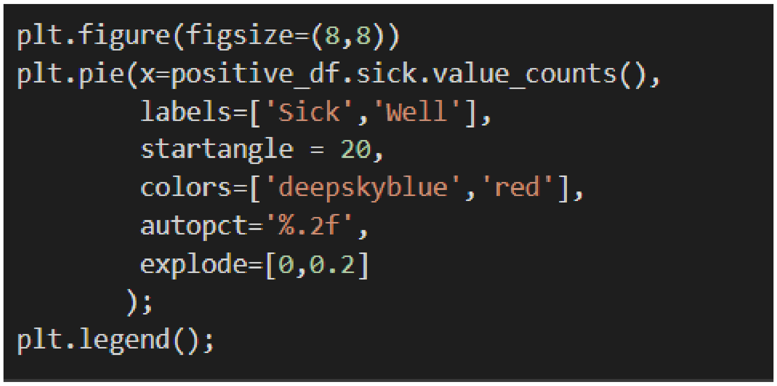
```
#matplotlib
import matplotlib.pyplot as plt #IMPORTING MATPLOTLIB

#seaborn
import seaborn as sns #IMPORTING SEABORN
```

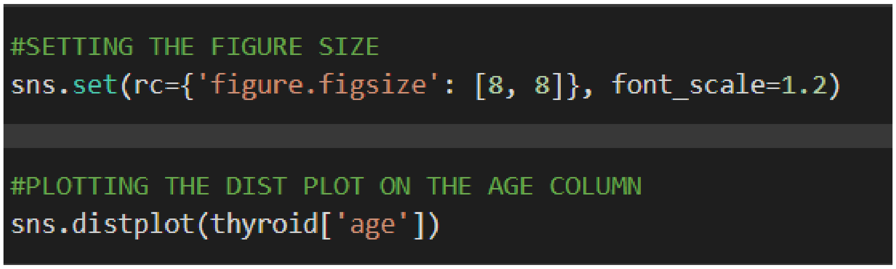
```
sns.countplot(x='binaryClass',data=thyroid) #PLOTING THE COUNTPLOT FOR THE TARGET VARIABLE
plt.title("Countplot for Target variable");
```



```
positive_df = thyroid[thyroid.binaryClass==1]
plt.figure(figsize=(9,6))
sns.histplot(x='age',data=positive_df,color='blue')
plt.title("Distribution of Positive Class Based on Age",{'fontsize':20});
```

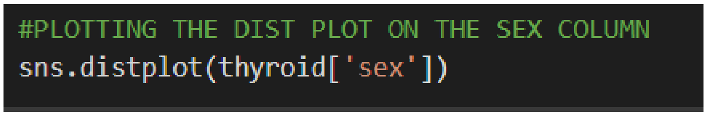


```
plt.figure(figsize=(8,8))
plt.pie(x=positive_df.sick.value_counts(),
        labels=['Sick','Well'],
        startangle = 20,
        colors=['deepskyblue','red'],
        autopct='%.2f',
        explode=[0,0.2]
       );
plt.legend();
```

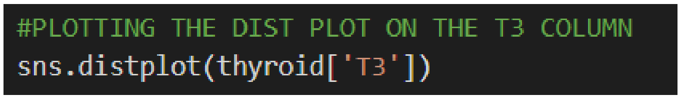


```
#SETTING THE FIGURE SIZE
sns.set(rc={'figure.figsize': [8, 8]}, font_scale=1.2)
```

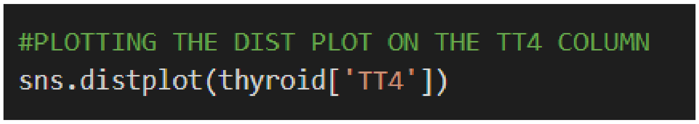
```
#PLOTTING THE DIST PLOT ON THE AGE COLUMN
sns.distplot(thyroid['age'])
```



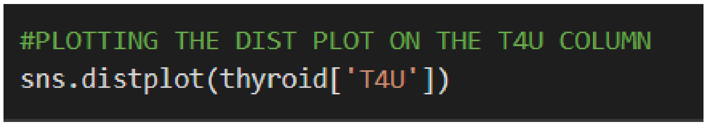
```
#PLOTTING THE DIST PLOT ON THE SEX COLUMN
sns.distplot(thyroid['sex'])
```



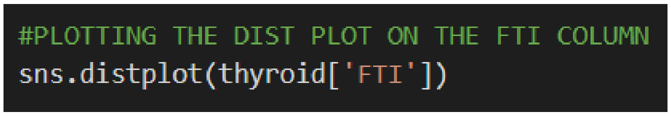
```
#PLOTTING THE DIST PLOT ON THE T3 COLUMN
sns.distplot(thyroid['T3'])
```



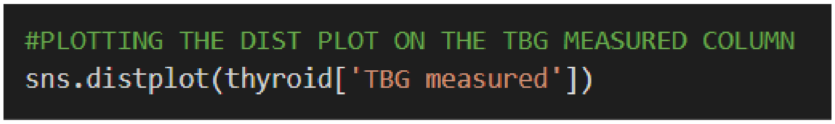
```
#PLOTTING THE DIST PLOT ON THE TT4 COLUMN
sns.distplot(thyroid['TT4'])
```



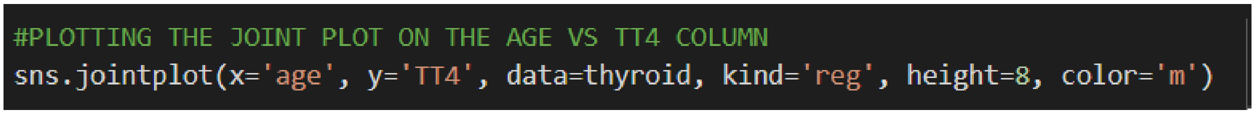
```
#PLOTTING THE DIST PLOT ON THE T4U COLUMN
sns.distplot(thyroid['T4U'])
```



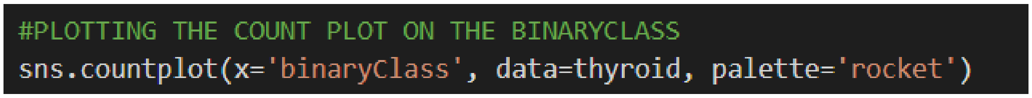
```
#PLOTTING THE DIST PLOT ON THE FTI COLUMN
sns.distplot(thyroid['FTI'])
```



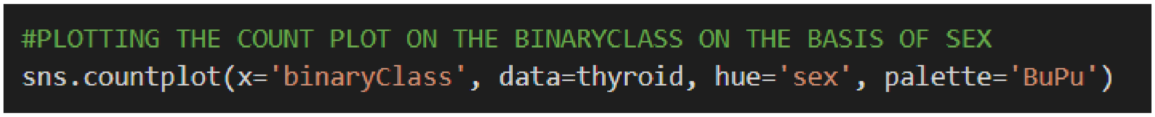
```
#PLOTTING THE DIST PLOT ON THE TBG MEASURED COLUMN
sns.distplot(thyroid['TBG measured'])
```



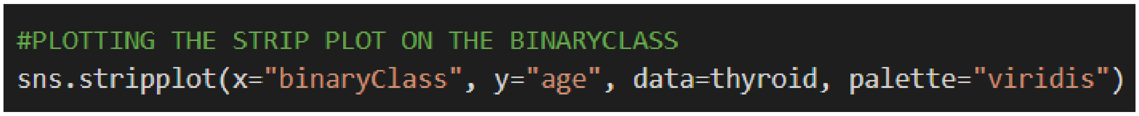
```
#PLOTTING THE JOINT PLOT ON THE AGE VS TT4 COLUMN
sns.jointplot(x='age', y='TT4', data=thyroid, kind='reg', height=8, color='m')
```



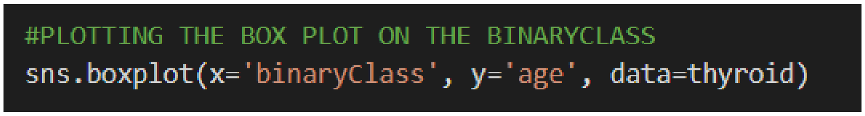
```
#PLOTTING THE COUNT PLOT ON THE BINARYCLASS
sns.countplot(x='binaryClass', data=thyroid, palette='rocket')
```



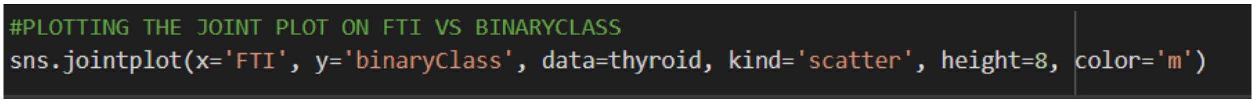
```
#PLOTTING THE COUNT PLOT ON THE BINARYCLASS ON THE BASIS OF SEX
sns.countplot(x='binaryClass', data=thyroid, hue='sex', palette='BuPu')
```



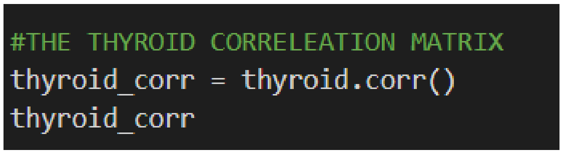
```
#PLOTTING THE STRIP PLOT ON THE BINARYCLASS
sns.stripplot(x="binaryClass", y="age", data=thyroid, palette="viridis")
```



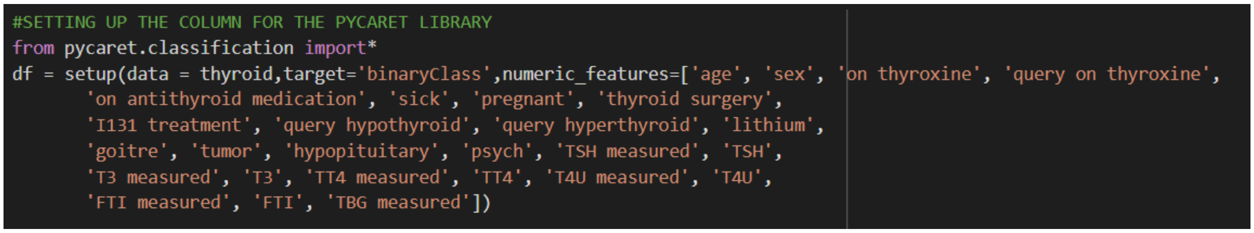
```
#PLOTTING THE BOX PLOT ON THE BINARYCLASS
sns.boxplot(x='binaryClass', y='age', data=thyroid)
```



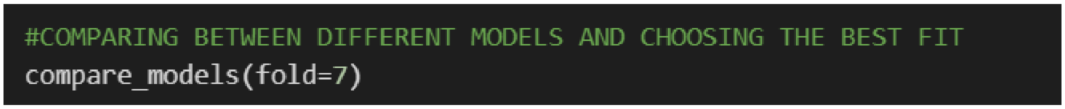
```
#PLOTTING THE JOINT PLOT ON FTI VS BINARYCLASS
sns.jointplot(x='FTI', y='binaryClass', data=thyroid, kind='scatter', height=8, color='m')
```



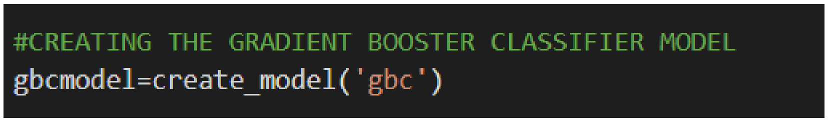
```
#THE THYROID CORRELEATION MATRIX
thyroid_corr = thyroid.corr()
thyroid_corr
```



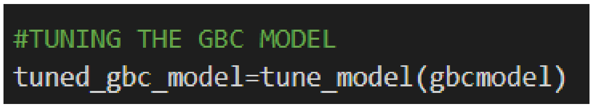
```
#SETTING UP THE COLUMN FOR THE PYCARET LIBRARY
from pycaret.classification import*
df = setup(data = thyroid,target='binaryClass',numeric_features=['age', 'sex', 'on thyroxine', 'query on thyroxine',
       'on antithyroid medication', 'sick', 'pregnant', 'thyroid surgery',
       'I131 treatment', 'query hypothyroid', 'query hyperthyroid', 'lithium',
       'goitre', 'tumor', 'hypopituitary', 'psych', 'TSH measured', 'TSH',
       'T3 measured', 'T3', 'TT4 measured', 'TT4', 'T4U measured', 'T4U',
       'FTI measured', 'FTI', 'TBG measured'])
```



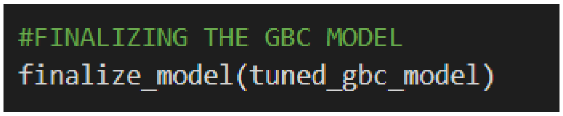
```
#COMPARING BETWEEN DIFFERENT MODELS AND CHOOSING THE BEST FIT
compare_models(fold=7)
```



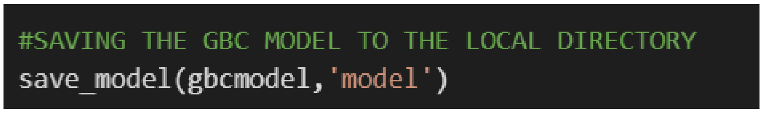
```
#CREATING THE GRADIENT BOOSTER CLASSIFIER MODEL
gbcmodel=create_model('gbc')
```



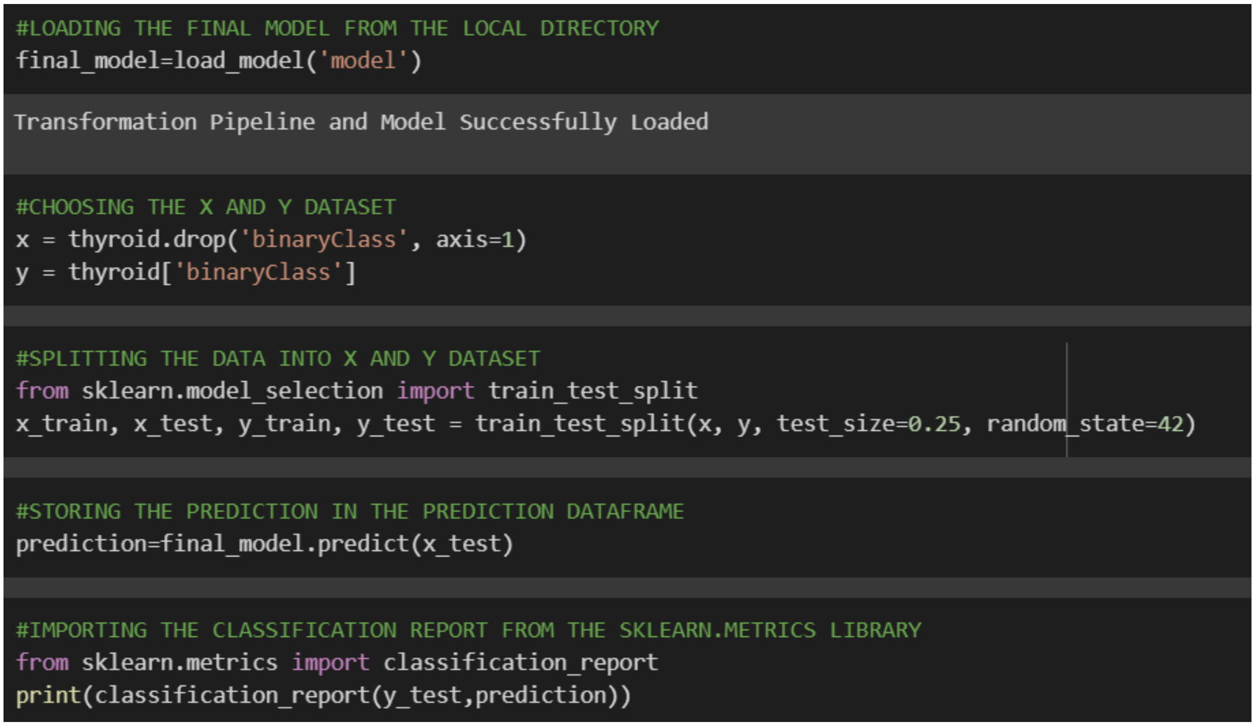
```
#TUNING THE GBC MODEL
tuned_gbc_model=tune_model(gbcmodel)
```



```
#FINALIZING THE GBC MODEL
finalize_model(tuned_gbc_model)
```



```
#SAVING THE GBC MODEL TO THE LOCAL DIRECTORY
save_model(gbcmodel,'model')
```



```
#LOADING THE FINAL MODEL FROM THE LOCAL DIRECTORY
final_model=load_model('model')
```

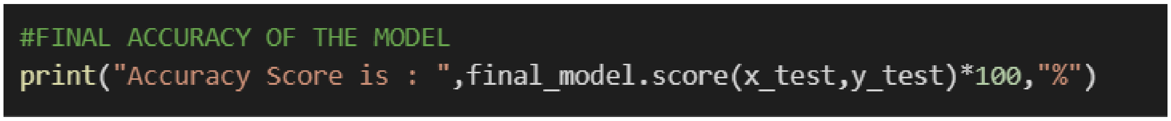
```
Transformation Pipeline and Model Successfully Loaded
```

```
#CHOOSING THE X AND Y DATASET
x = thyroid.drop('binaryClass', axis=1)
y = thyroid['binaryClass']
```

```
#SPLITTING THE DATA INTO X AND Y DATASET
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.25, random_state=42)
```

```
#STORING THE PREDICTION IN THE PREDICTION DATAFRAME
prediction=final_model.predict(x_test)
```

```
#IMPORTING THE CLASSIFICATION REPORT FROM THE SKLEARN.METRICS LIBRARY
from sklearn.metrics import classification_report
print(classification_report(y_test,prediction))
```



```
#FINAL ACCURACY OF THE MODEL
print("Accuracy Score is : ",final_model.score(x_test,y_test)*100,"%")
```