LONDON
METROPOLITAN
UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CU6051NI – Artificial Intelligence

Assessment Weightage & Type

75% Individual Coursework

2023-24 Spring

Student Name: KATYANI BAJGAIN

Group: L3C7

College ID: NP01CP4A210175

London Met ID: 21049520

Assignment Submission Date: Tuesday, January 17, 2023

Submitted To: Mr. Rajesh Mahae

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a mark of zero will be awarded.

Table of Contents

Table of Figures

Table of tables

1. Introduction

Artificial Intelligence (AI) refers to the simulation of human intelligence processes by machines especially computer systems. These processes include learning, reasoning, problem-solving, perception, and language understanding (B.J.Copeland, 2024).

AI can be categorized into two types: Weak Ai and Strong AI. Weak Ai, also known as narrow AI, is designed to perform a specific task. It operates within a limited context and is simulation of human intelligence applied to a narrowly defined problem, such as driving a car, transcribing human speech, or curating content on a website. Examples of Weak AI include Siri, Alexa, self-driving cars, Google Search, conversational bots, email spam filters, and Netflix’s recommendations (Schroer, 2024).

On the other hand, Strong AI, also known as Artificial General Intelligence (AGI), is a type of AI that can replicate the cognitive abilities of the human brain. Unlike Weak AI, which is designed to perform a specific task, Strong Ai can understand, learn, and apply knowledge across various domains. It can solve problems autonomously and adapt to new situations. In theory, a strong AI system should be able to pass both a Turing test and the Chinese Room argument (laskowski, 2024).

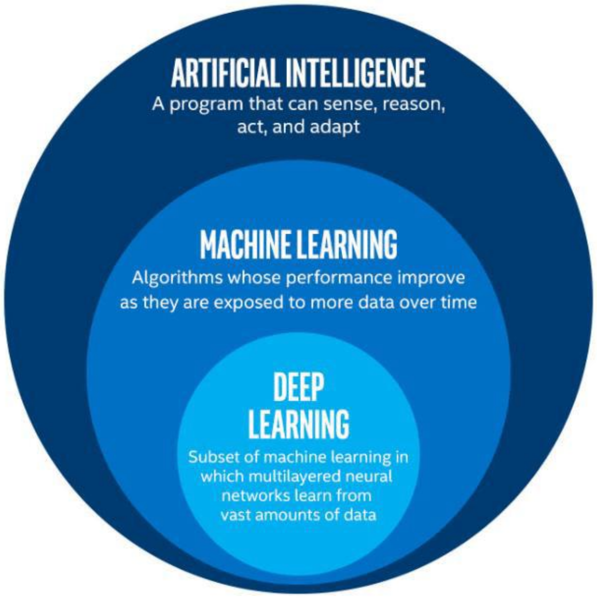

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

Figure 1:Core concepts of AI (Oliviera, 2019)

1.1. Explanation of AI concepts

Artificial Intelligence is a fascinating field that is deeply embedded in our daily lives. It's a branch of computer science focused on creating intelligent machines that can perform tasks that usually require human intelligence. This can include learning from data, recognizing patterns, making decisions, and even understanding natural language (seidor , 2021). Some key components of AI are mentioned below:

- **Machine learning:** One of the key components of AI is Machine Learning (ML). ML is a method of data analysis that automates the building of analytical models. It's a type of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. For example, Machine Learning algorithms can analyse historical weather data and learn to predict future weather patterns, or they can learn from past instances of spam and legitimate emails to filter out spam (seidor , 2021).

- **Deep Learning:** Deep Learning is a subset of ML that uses neural networks with many layers to model and understand complex patterns. These neural networks attempt to simulate the behaviour of the human brain – albeit far from matching its ability – so they can "learn" from vast amounts of data. Deep learning is commonly used in image recognition, where algorithms can distinguish between different objects in images. For instance, a deep learning algorithm can be trained to recognize faces in photos, or differentiate between images of dogs and cats (seidor , 2021).

1. **Natural Language Processing (NPL):** NPL is another important aspect of AI. NPL is a field of AI that gives the machines the ability to read, understand, and derive meaning from human languages. For example, NPL can be used in sentiment analysis, where it determines whether a piece of writing is positive, negative, or neutral. An example of this is Twitter's sentiment analysis tool, which uses NLP to gauge public opinion on specific topics. (seidor , 2021)

Another application of NLP is in machine translation, where it translates text from one language to another. Google Translate is a prime example of this, using NLP to translates text from one language to another. In conclusion, AI, ML, DL, NLP are integral parts of modern computing. They have wide-ranging applications in various sectors, from healthcare to finance, transportation, and more, and continue to evolve as research progress.

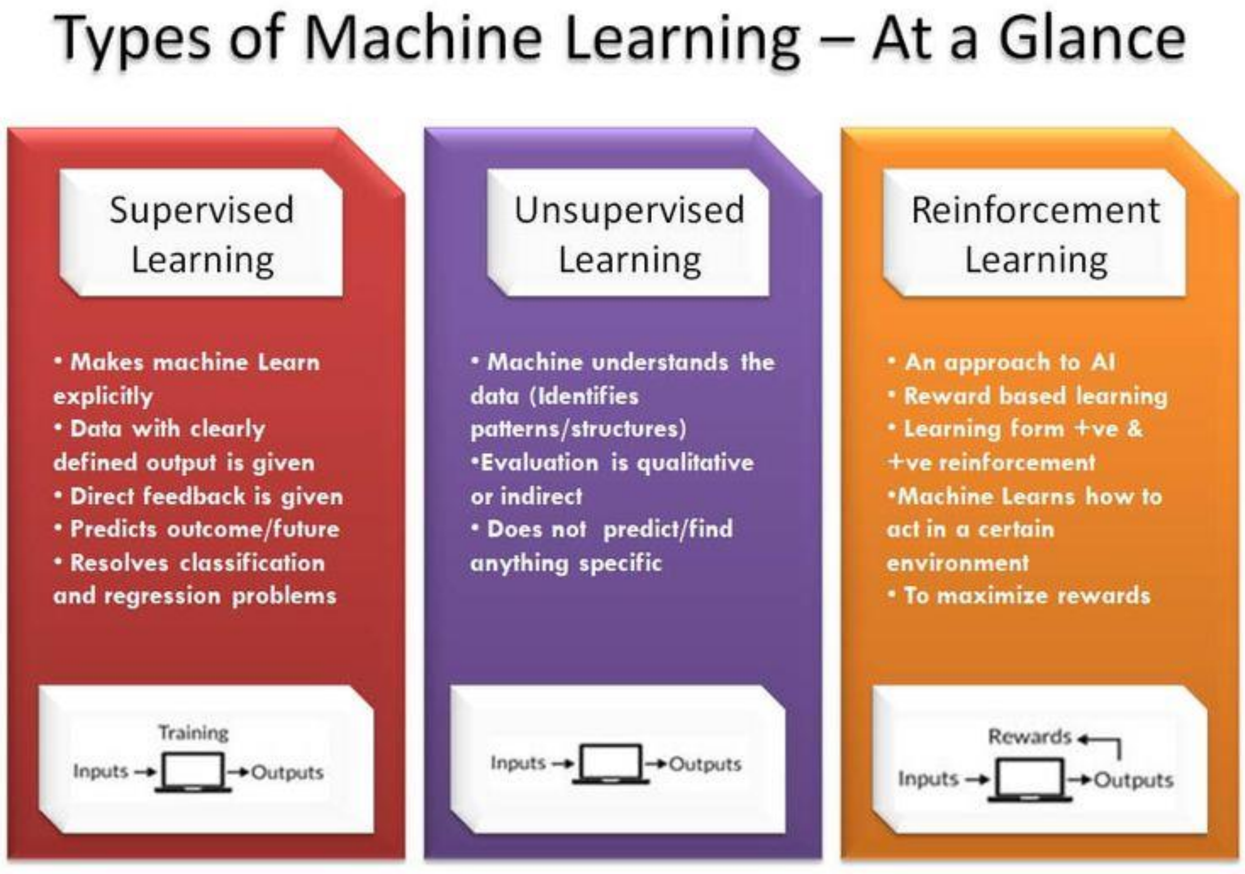
1.1.1. Three categories of Machine Learning are listed below:

Machine Learning (ML) is a subset of Artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. It primarily involves the creation of algorithms that allow computers to learn automatically (Sarker, 2021). There are three main types of Machine learning techniques: Supervised learning, Unsupervised Learning, Reinforcement Learning.

2. **Supervised machine learning** – Supervised learning equips machines with labelled input-output data to create a mapping function for predicting outputs. It continues training until reaching the desired accuracy level. This method aims to teach machines classification systems, like Gmail’s spam filtering by google. Common algorithms in this category include Nearest Neighbour, Naïve Bayes, Decision Trees, Random Forest and Neural networks. (Sodhi, et al., 2019)

3. **Unsupervised machine learning** – In unsupervised learning, machines are given unlabelled and unclassified input datasets. The algorithm then develops a function to uncover concealed structure within the datasets based on patterns, similarities, and differences among the data without prior training. Unlike supervised learning, there’s no evaluation of the identified structure’s accuracy. Unsupervised learning often revolves around solving clustering and association problems. Prominent algorithms in this domain include the k-means algorithm for clustering and the A prior algorithm for association problems. (Sodhi, et al., 2019)
4. **Reinforcement learning** – The machine operates in an environment where it learns through trial and error, making decisions and acquiring knowledge from its actions

and prior experiences. When then machine makes a correct decision, it receives a reward signal from the environment, reinforcing the successful action, and stores information about the rewarded state-action pair. Subsequently, the machine replicates the rewarded behaviour when encountering similar situations. Reinforcement learning algorithms find application in domains emphasizing strategic decision-making, such as in Self-Driving Cars. Among the prevalent reinforcement learning algorithms are Q-learning and Markov Decisions Process (Sodhi, et al., 2019)



Types of Machine Learning – At a Glance

Supervised Learning

- Makes machine Learn explicitly
- Data with clearly defined output is given
- Direct feedback is given
- Predicts outcome/future
- Resolves classification and regression problems

Unsupervised Learning

- Machine understands the data (Identifies patterns/structures)
- •Evaluation is qualitative or indirect
- Does not predict/find anything specific

Reinforcement Learning

- An approach to AI
- Reward based learning
- Learning form +ve & +ve reinforcement
- •Machine Learns how to act in a certain environment
- To maximize rewards

1.1.2. Supervised Learning

Supervised Learning is a subcategory of Machine Learning and Artificial Intelligence. It is characterized by its use of labelled datasets to train algorithms to classify data or predict outcome accurately. In Supervised Learning, a machine is trained using 'labelled' data. Datasets are said to be labelled when they contain both input and output parameters. In other words, the data has already been tagged with the correct answer. Supervised learning uses a training set to teach models to yield the desired output. This training dataset includes inputs and correct outputs, which allow the model to learn over time. The algorithm measures its accuracy through the loss function, adjusting until the error has been sufficiently minimized. (IBM, 2024). Supervised learning can be divided into two types of problems when data mining – classification and regression.

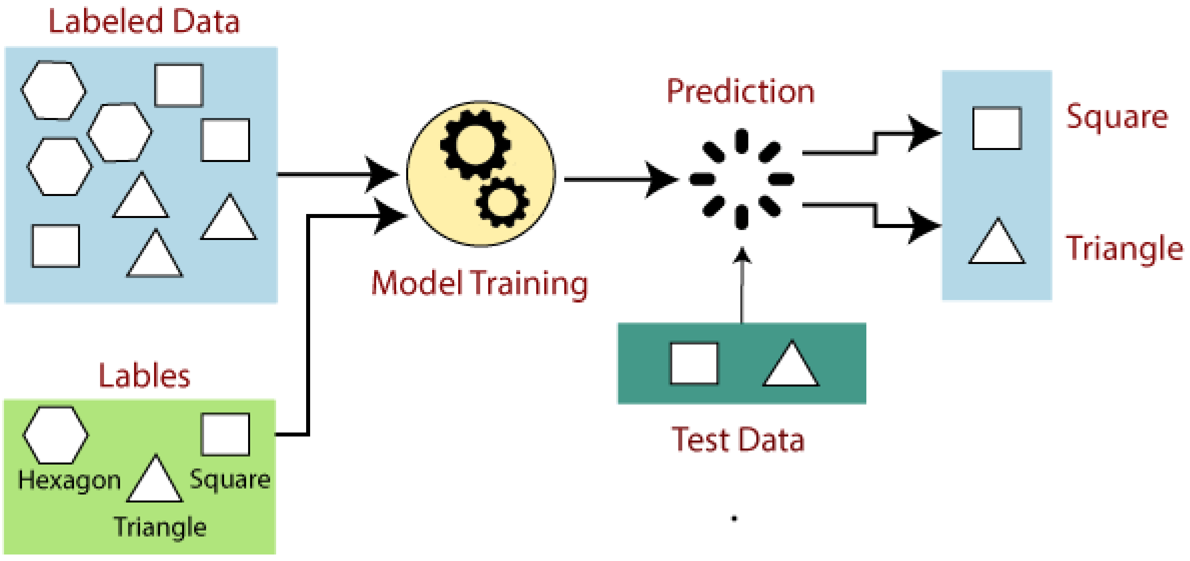


Figure 2:Supervised Learning (javatpoint, 2024)

- **Classification:** This uses an algorithm to accurately assign test data into specific categories. It recognizes specific entities within the dataset and attempts to draw some conclusions on how those entities should be labelled or defined. Common classification algorithms are linear classifiers, support vector machines (SVM), decision trees, k-nearest neighbour, and random forest (IBM, 2024).

- **Regression:** This is used to understand the relationship between dependent and independent variables. It is commonly used to make projections, such as for sales revenue for a given business. Linear regression, logistical regression, and polynomial regression are popular regression algorithms (IBM, 2024).

Despite its advantages, supervised learning does have its limitations. Concrete examples are required for training classifiers, and decision boundaries can be overtrained in the absence of the right examples. Additionally, difficulties may arise when classifying big data.

1.1.3. Unsupervised Learning

Unsupervised Learning is a subset of Machine Learning where the algorithm learns from unlabelled data without any predefined outputs or target variables. The purpose of unsupervised learning is to find patterns, similarities, or groupings within the data to gain insights and make data-driven decisions. (DatabaseTown, 2024)

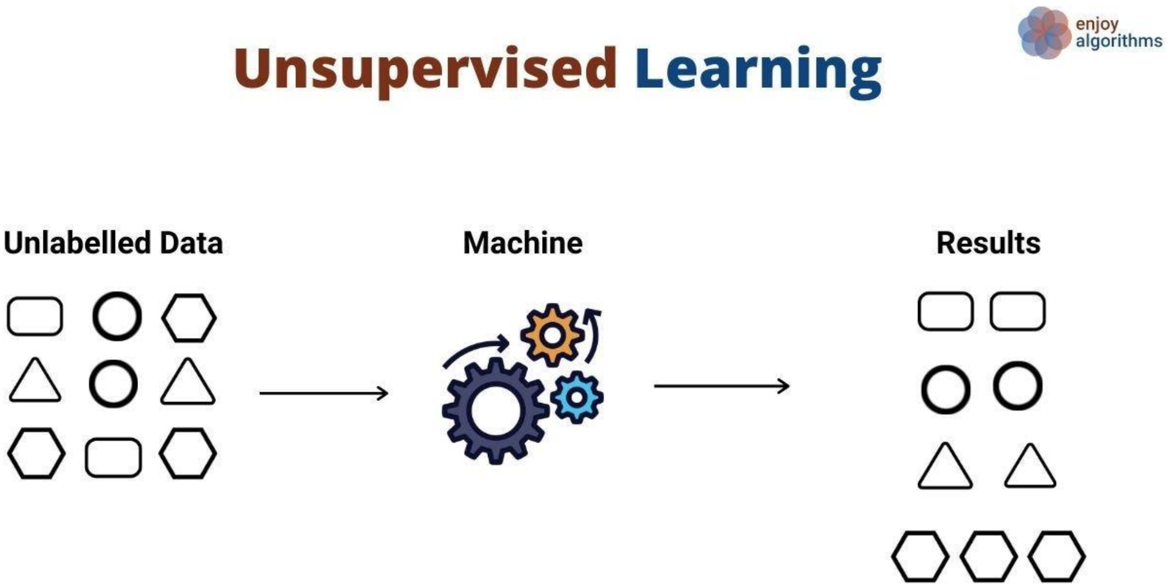


Figure 3:Unsupervised Learning (enjoy algorithms, 2024)

This type of learning is particularly useful when dealing with large datasets where manual labelling would be impractical or costly. It helps us find hidden patterns or structures in data that doesn’t have any labels. It provides valuable insights and knowledge by uncovering meaningful connections and information that might not have been noticed before. Unsupervised learning is beneficial for data science teams who aren't sure what they're seeking in the data. It can be used to search for unknown similarities and differences in data and create corresponding groups. For instance, user categorization based on their social media activity. The method doesn't require training data to be labelled, saving time spent on manual classification tasks. Such an approach can find unknown patterns and therefore useful insights in data that couldn’t be found otherwise. It also reduces the chance of human error and bias, which could occur during manual labelling processes. (altexoft, 2021)

However, despite its advantages, unsupervised learning has some limitations and challenges. Since unsupervised learning deals with unlabelled data, there is no definitive measure of correctness or accuracy. Evaluation and interpretation of results become subjective and rely heavily on domain expertise. Unsupervised learning algorithms often provide clusters or patterns without explicit labels or explanations. Interpreting and understanding the meaning of these clusters can be challenging and subjective. Unlike supervised learning, where the algorithm learns from explicit feedback, unsupervised learning lacks explicit guidance, which can result in the algorithm discovering irrelevant or noisy patterns. (DatabaseTown, 2024)

1.1.4. Reinforcement Learning

Reinforcement Learning (RL) is a subfield of machine learning where an agent learns to make decisions by interacting with an environment. The agent's goal is to learn a policy, which is a strategy that specifies what action to take under what circumstances. The policy is learned by the agent observing the consequences of its actions and adjusting its behaviour accordingly. In RL, the agent receives feedback in the form of rewards or penalties after each action it performs. The agent's goal is to learn a policy that maximizes the total amount of reward it receives over time. This is achieved by repeatedly interacting with the environment, learning from the feedback it receives, and updating its policy based on this feedback Reinforcement Learning has found applications in various fields such as marketing and advertising, gaming, health care and manufacturing. (Santa Clara University , 2024)

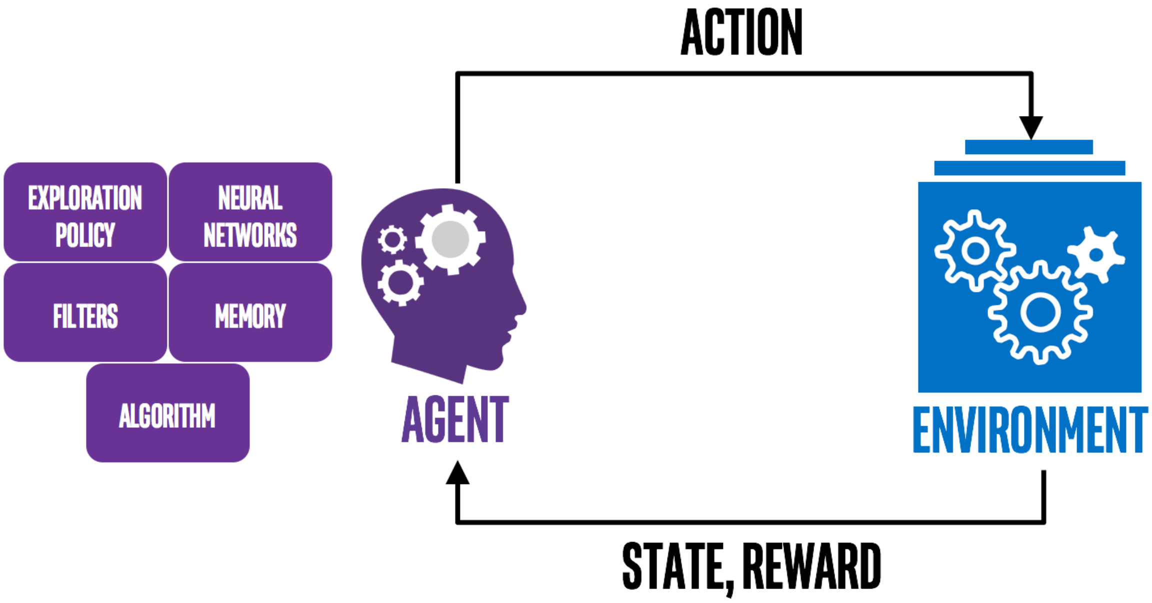


Figure 4:Reinforcemnet Learning

Despite its potential, RL also faces several challenges. One of the main challenges is the difficulty of defining a suitable reward function. Another challenge is the computational complexity of RL algorithms, especially in high-dimensional spaces. Finally, RL algorithms often require a large amount of data to learn effectively, which can be expensive to collect.

1.2. Problem Domain

Breast cancer is a prevalent health issue, particularly among women globally, with two primary forms: benign (non-cancerous) and malignant (cancerous). Studies by the World Cancer Research Fund International have indicated a growing trend in breast cancer incidences, particularly in the Asia/Pacific region, at a rate of 1.7% annually. The early detection of breast cancer is a critical factor in improving patient outcomes. Despite the high mortality rate associated with breast cancer, early diagnosis significantly enhances the chances of successful treatment and survival. Thus, accurate classification of cancer types has emerged as a critical requirement within the field of cancer research. (Islan, et al., 2019)

Malignant tumours pose a greater threat due to their rapid growth rate compared to benign tumours. Therefore, the early identification of tumour types is of utmost importance for providing appropriate treatment for breast cancer patients. With the availability of comprehensive datasets and precise classifiers, it might be possible to develop an automated diagnostic system capable of assisting in the preliminary diagnosis of breast cancer. Such a system could not only enhance the precision of diagnoses but also reduce the workload on healthcare professionals, thereby improving the overall efficiency of the healthcare system.

Due to breast cancer, a number of women die every year. With an early diagnosis, breast cancer can be cured. Prognosis and early detection of cancer types have become a necessity in cancer research. Thus, a reliable and accurate system is required for the classification of benign and malignant tumour types of breast cancer (Islan, et al., 2019). Malignant tumours are deadly as their rate of growth is much higher than benign tumours. So, early identification of tumour type is pivotal for the appropriate treatment of a patient having breast cancer. (Sharmin & Das Annesha, 2021) with the help of a proper dataset and accurate classifiers an automatic diagnostic system for preliminary diagnosis of breast cancer can be built.

2. Background

In the field of contemporary healthcare, the incorporation of innovative technologies has dramatically altered disease detection and prognosis. Among these advancements, the Breast Cancer Prediction System stands out. This system is an intricate application that employs supervised machine learning algorithms to heighten the precision of breast cancer predictions.

This ground-breaking development is built upon the foundational data of the Wisconsin Breast Cancer (WBCD) dataset. The WBCD dataset, known for its extensive and carefully compiled information, forms the basis for training and refining the machine learning model. The WBCD dataset includes various attributes derived from digitized images of fine needle aspirates (FNA) of breast masses. This wealth of information serves as a rich source for the algorithm to learn and make predictions. By integrating cutting-edge technology with a robust dataset, the Breast Cancer Prediction System is aimed at facilitating early detection and improved prognosis. This could potentially save lives and elevate the quality of healthcare for individuals at risk.

In summary, the Breast Cancer Prediction System represents a significant leap forward in predictive analytics for breast cancer. It combines advanced technology with a comprehensive dataset to enhance the accuracy of breast cancer predictions. The system's ultimate goal is to aid in early detection and improved prognosis, thereby potentially saving lives and enhancing the quality of healthcare for individuals at risk.

2.1. Research work done on the chosen topic/problem domain.

2.1.1. Breast Cancer

Breast cancer is a serious condition that can be either benign or malignant. Benign breast tumours are noncancerous growths that occur in the breasts. They usually don't cause any symptoms and don't require treatment unless they become large enough to cause discomfort or pain. In many cases, they can simply be left alone. However, benign tumours can sometimes cause complications, such as pain or mammary duct obstruction. On the other hand, malignant breast tumours are cancerous (National Breast Cancer Foundation, 2024).

This means they can grow and spread to other parts of the body. Malignant tumours are divided into several types, including ductal carcinoma in situ (DCIS), invasive ductal carcinoma, and lobular carcinoma. These tumours can be life-threatening and require immediate medical attention. The treatment for breast cancer depends on the type of tumour and its stage. It may involve surgery, radiation therapy, chemotherapy, hormone therapy, targeted therapy, immunotherapy, or a combination of these treatments. The goal of treatment is to kill the cancer cells, control the disease, and relieve symptoms.

In addition to treatment, lifestyle changes such as regular exercise, a healthy diet, avoiding alcohol, and quitting smoking can also help reduce the risk of developing breast cancer. Regular screening tests, like mammograms, can also detect breast cancer early when it's still small and less likely to have spread. (Cleveland Clinic, 2024)

2.1.2. Prediction Systems using machine learning

Machine Learning (ML) and Artificial Intelligence (AI) have revolutionized many industries by enabling accurate predictions based on historical data. These predictions are crucial for decision-making, strategy planning, and resource allocation.

ML is a subset of AI that allows computers to learn from data and improve their performance over time. It can recognize patterns in data and make predictions based on these patterns. For instance, given a set of features describing a person, an ML model can predict whether that person is ill or healthy. Similarly, given a set of parameters describing an animal, an ML model can predict whether that animal is being treated or under control. (Bo, et al., 2023). ML is used in various industries, including healthcare, finance, retail, manufacturing, automotive, and technology. For example, in the medical field, predictive algorithms analyse patient data to aid healthcare providers in making more accurate and timely decisions, thereby improving patient care while streamlining operational processes. (Seraydarian, 2023)In financial sector, ML can be used to predict stock prices, credit risk. And customer behaviour. IN retail, ML can help predict sales, inventory needs, and customer preferences. In manufacturing, ML can predict vehicle failure, optimize fuel consumption, and enhance driver safety. (Seraydarian, 2023)

In conclusion ML and AI have the potential to transform traditional practices by providing versatile applications that bring new efficiencies, insights, and transformation. Their adaptability makes them a valuable asset for business across various sectors, reshaping traditional practices and enabling more intelligent decision-making and automation. (Seraydarian, 2023)

2.1.3. Breast Cancer prediction using ML and its real-world case studies

Breast cancer prediction using machine learning has shown promising results in recent years. Machine learning models can analyse vast amounts of data to identify patterns and make predictions, improving the accuracy of diagnoses and treatment plans. A notable example of this approach is the Breast Cancer Prediction System, which uses the Wisconsin Breast Cancer (WBCD) dataset to train its models. (Reza, et al., 2022)

One real-world case study involved a research project where 5178 records of people were analysed retrospectively. Each record contained 24 features, including demographic, laboratory, and mammography features. The model was trained to recognize the presence or absence of breast cancer based on these features. The study concluded that the proposed machine-learning approaches could predict breast cancer, allowing for early detection and potentially slowing down the progress of the disease, thus reducing mortality rates. (Reza, et al., 2022)

Another study demonstrated the effectiveness of machine learning in breast cancer detection and prevention. The researchers applied different machine learning approaches to larger datasets from different institutions, considering key features from a variety of relevant data sources. This multi-centre study showed that applying machine learning techniques to breast cancer prediction could improve the performance of modelling and enable targeted prevention and treatment strategies. (Arslan, et al., 2023)

These case studies underscore the potential of machine learning in breast cancer prediction. However, it's important to remember that machine learning models are only as good as the data they're trained on. Therefore, collecting and analysing large, comprehensive datasets remains a key challenge in this field.

2.1.4. Advantages of Machine Learning in Breast Cancer Prediction

Machine learning offers a plethora of advantages for breast cancer prediction, significantly advancing the fight against this prevalent disease. Here are some key benefits:

1. **Early Detection:**
    - **Enhanced Pattern Recognition**: ML algorithms excel at identifying subtle patterns and complex relationships within vast datasets, potentially revealing early-stage cancer indicators invisible to the human eye.
    - **Personalized Risk Assessment:** By analysing individual patient data, ML models can predict cancer risk with greater accuracy, enabling proactive interventions and early screenings for high-risk individuals.
2. **Improved Accuracy and Efficiency:**
    - **Reduced False Positives and Negatives:** ML algorithms can refine diagnostic approaches, minimizing unnecessary biopsies and emotional distress caused by false positives, while maximizing early detection through reduced false negatives.
    - **Faster and More Accessible Diagnosis:** Automated analysis techniques, powered by ML, can expedite diagnostic workflows, potentially increasing diagnostic capacity and accessibility in resource-constrained settings.
3. **Tailored Treatment and Prognosis:**
    - **Risk Stratification and Subtype Identification:** ML models can help classify breast cancer subtypes and predict tumour aggressiveness, guiding personalized treatment plans and resource allocation based on individual risk profiles.
    - **Improved Treatment Response Prediction:** By analysing response data from past patients, ML algorithms can estimate the effectiveness of different treatment options for individual cases, optimizing treatment strategies and increasing success rates.

4. **Continuous Learning and Improvement:**
    - **Adaptability to New Data:** Unlike static models, ML algorithms can continuously learn and adapt to new data, incorporating ongoing research findings and updating predictions over time, leading to ever-increasing accuracy and effectiveness.
    - **Scalability and Generalizability:** Well-trained ML models can be applied to wider populations, potentially impacting healthcare systems globally and contributing to advancements in public health interventions and resource allocation.

Overall, machine learning offers a powerful toolkit for revolutionizing breast cancer prediction and diagnosis. By empowering early detection, personalized treatment, and continuous improvement, it holds immense potential to save lives and improve the well-being of countless individuals impacted by this disease.

2.1.5. Disadvantages of Machine Learning in Breast Cancer Prediction

While machine learning (ML) offers immense potential in breast cancer prediction, it's crucial to acknowledge its limitations and potential downsides. Here are some important considerations:

- **Data Bias:** ML algorithms learn from the data they're trained on. If the data is biased (e.g., lacking diversity in demographics or clinical presentations), the resulting model can reflect and perpetuate these biases, leading to inaccurate predictions for certain groups.

- **Algorithm Explain ability:** Complex ML models can be difficult to interpret, making it challenging to understand how they arrive at their predictions. This lack of transparency can hinder trust and confidence in the technology, particularly for healthcare professionals.

- **False Positives and Negatives:** Even well-trained models can generate false positives or negatives. False positives can cause anxiety and unnecessary procedures, while false negatives can delay diagnosis and treatment.

- **Generalizability:** Models trained on specific datasets may not generalize well to different populations or healthcare settings. This can limit their practical application in diverse clinical contexts.

- **Overreliance on Technology:** ML should be seen as a tool to enhance, not replace, clinical expertise. Overreliance on predictions without considering a patient's full medical picture can lead to misdiagnosis or inappropriate treatment.

- **Technical Infrastructure and Expertise:** Implementing and maintaining ML models requires significant technical infrastructure and expertise, which may not be readily available in all healthcare settings.

- **Ethical Considerations**: Issues like data privacy, algorithmic bias, and potential discrimination against high-risk individuals need careful consideration and ethical frameworks to guide the development and use of ML in healthcare.

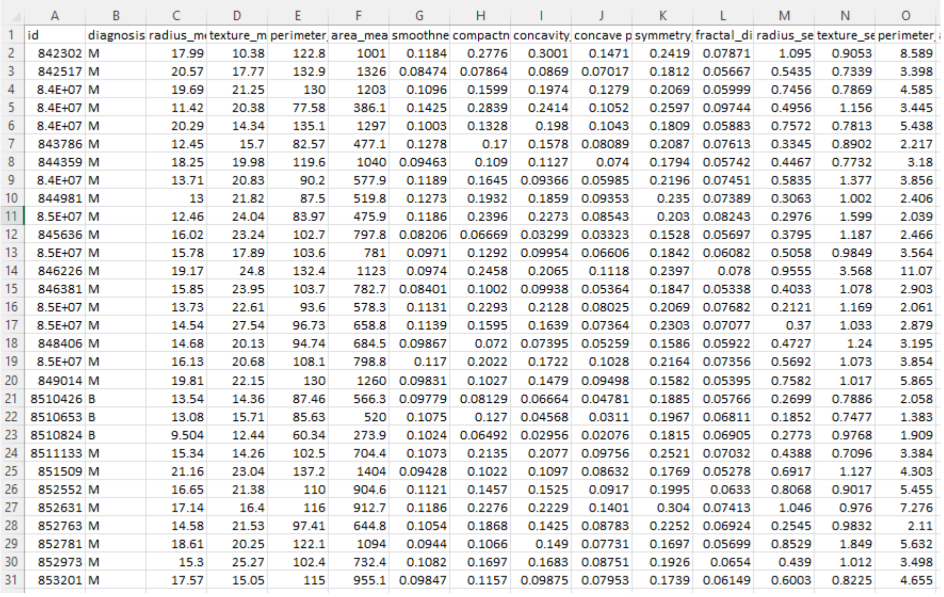
Despite these challenges, the potential benefits of ML in breast cancer prediction are undeniable. By developing reliable models, addressing biases, and ensuring responsible use, this technology can contribute significantly to improved early detection and personalized cancer care.

2.1.6. Explanation of the Dataset

The Breast Cancer Wisconsin (Diagnostic) Dataset is a collection of data related to breast cancer diagnosis, created by Dr William H. Wolberg at the University of Wisconsin Hospital. The dataset includes %^(records, each with 31 columns, and is used in research to apply machine learning techniques to medical diagnosis. It provides an opportunity to study the characteristic of the given dataset, commonly called Exploratory Data Analysis (EDA). The Random Forest algorithm is used in this study to analyse the medical case diagnosis of breast cancer. The Random Forest algorithm can combine the characteristics of multiple eigenvalues, and the combined results of multiple decision trees can be used to improve the prediction accuracy. The attributes information of the data set is presented below:

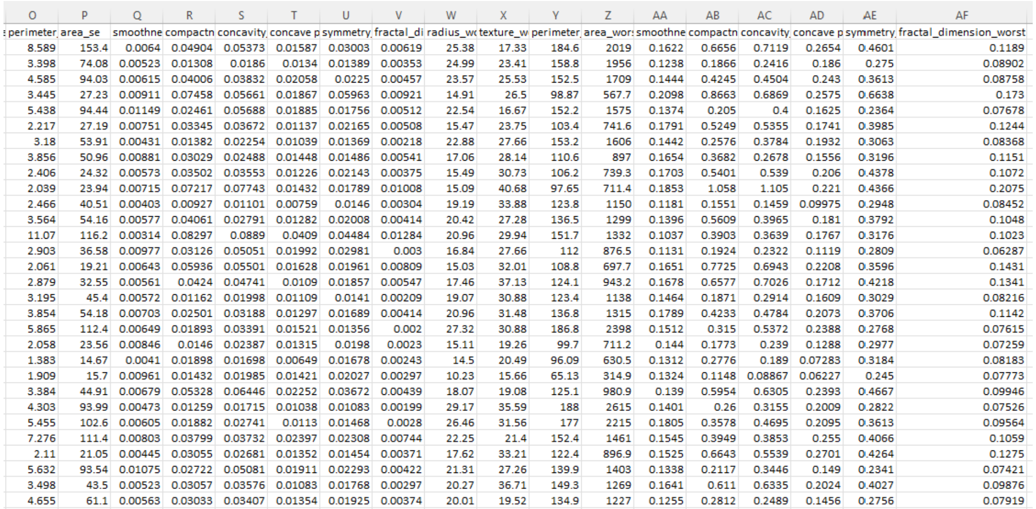
- ID number
- Diagnosis (M = malignant, B = benign)
- Ten real-valued features are computed for each cell nucleus:
    - radius (mean of distances from centre to points on the perimeter)
    - texture (standard deviation of grey-scale values)
    - perimeter
    - area
    - smoothness (local variation in radius lengths)
    - compactness (perimeter^2 / area - 1.0)
    - concavity (severity of concave portions of the contour)
    - concave points (number of concave portions of the contour)
    - concave points (number of concave portions of the contour)
    - fractal dimension ("coastline approximation" - 1)
- The mean, standard error and "worst" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius. All feature values are recoded with four significant digits.
- Class Distribution: 375 benign, 212 Malignant

The pictures of the dataset is presented below:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	id	diagnosis	radius_m	texture_m	perimeter_	area_mea	smoothne	compactn	concavity_	concave p	symmetry_	fractal_di	radius_se	texture_se	perimeter_
2	842302	M	17.99	10.38	122.8	1001	0.1184	0.2776	0.3001	0.1471	0.2419	0.07871	1.095	0.9053	8.589
3	842517	M	20.57	17.77	132.9	1326	0.08474	0.07864	0.0869	0.07017	0.1812	0.05667	0.5435	0.7339	3.398
4	8.4E+07	M	19.69	21.25	130	1203	0.1096	0.1599	0.1974	0.1279	0.2069	0.05999	0.7456	0.7869	4.585
5	8.4E+07	M	11.42	20.38	77.58	386.1	0.1425	0.2839	0.2414	0.1052	0.2597	0.09744	0.4956	1.156	3.445
6	8.4E+07	M	20.29	14.34	135.1	1297	0.1003	0.1328	0.198	0.1043	0.1809	0.05883	0.7572	0.7813	5.438
7	843786	M	12.45	15.7	82.57	477.1	0.1278	0.17	0.1578	0.08089	0.2087	0.07613	0.3345	0.8902	2.217
8	844359	M	18.25	19.98	119.6	1040	0.09463	0.109	0.1127	0.074	0.1794	0.05742	0.4467	0.7732	3.18
9	8.4E+07	M	13.71	20.83	90.2	577.9	0.1189	0.1645	0.09366	0.05985	0.2196	0.07451	0.5835	1.377	3.856
10	844981	M	13	21.82	87.5	519.8	0.1273	0.1932	0.1859	0.09353	0.235	0.07389	0.3063	1.002	2.406
11	8.5E+07	M	12.46	24.04	83.97	475.9	0.1186	0.2396	0.2273	0.08543	0.203	0.08243	0.2976	1.599	2.039
12	845636	M	16.02	23.24	102.7	797.8	0.08206	0.06669	0.03299	0.03323	0.1528	0.05697	0.3795	1.187	2.466
13	8.5E+07	M	15.78	17.89	103.6	781	0.0971	0.1292	0.09954	0.06606	0.1842	0.06082	0.5058	0.9849	3.564
14	846226	M	19.17	24.8	132.4	1123	0.0974	0.2458	0.2065	0.1118	0.2397	0.078	0.9555	3.568	11.07
15	846381	M	15.85	23.95	103.7	782.7	0.08401	0.1002	0.09938	0.05364	0.1847	0.05338	0.4033	1.078	2.903
16	8.5E+07	M	13.73	22.61	93.6	578.3	0.1131	0.2293	0.2128	0.08025	0.2069	0.07682	0.2121	1.169	2.061
17	8.5E+07	M	14.54	27.54	96.73	658.8	0.1139	0.1595	0.1639	0.07364	0.2303	0.07077	0.37	1.033	2.879
18	848406	M	14.68	20.13	94.74	684.5	0.09867	0.072	0.07395	0.05259	0.1586	0.05922	0.4727	1.24	3.195
19	8.5E+07	M	16.13	20.68	108.1	798.8	0.117	0.2022	0.1722	0.1028	0.2164	0.07356	0.5692	1.073	3.854
20	849014	M	19.81	22.15	130	1260	0.09831	0.1027	0.1479	0.09498	0.1582	0.05395	0.7582	1.017	5.865
21	8510426	B	13.54	14.36	87.46	566.3	0.09779	0.08129	0.06664	0.04781	0.1885	0.05766	0.2699	0.7886	2.058
22	8510653	B	13.08	15.71	85.63	520	0.1075	0.127	0.04568	0.0311	0.1967	0.06811	0.1852	0.7477	1.383
23	8510824	B	9.504	12.44	60.34	273.9	0.1024	0.06492	0.02956	0.02076	0.1815	0.06905	0.2773	0.9768	1.909
24	8511133	M	15.34	14.26	102.5	704.4	0.1073	0.2135	0.2077	0.09756	0.2521	0.07032	0.4388	0.7096	3.384
25	851509	M	21.16	23.04	137.2	1404	0.09428	0.1022	0.1097	0.08632	0.1769	0.05278	0.6917	1.127	4.303
26	852552	M	16.65	21.38	110	904.6	0.1121	0.1457	0.1525	0.0917	0.1995	0.0633	0.8068	0.9017	5.455
27	852631	M	17.14	16.4	116	912.7	0.1186	0.2276	0.2229	0.1401	0.304	0.07413	1.046	0.976	7.276
28	852763	M	14.58	21.53	97.41	644.8	0.1054	0.1868	0.1425	0.08783	0.2252	0.06924	0.2545	0.9832	2.11
29	852781	M	18.61	20.25	122.1	1094	0.0944	0.1066	0.149	0.07731	0.1697	0.05699	0.8529	1.849	5.632
30	852973	M	15.3	25.27	102.4	732.4	0.1082	0.1697	0.1683	0.08751	0.1926	0.0654	0.439	1.012	3.498
31	853201	M	17.57	15.05	115	955.1	0.09847	0.1157	0.09875	0.07953	0.1739	0.06149	0.6003	0.8225	4.655

Figure 5:Dataset Example 1



O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF
perimeter_	area_se	smoothne	compactn	concavity_	concave p	symmetry_	fractal_di	radius_wo	texture_w	perimeter_	area_wor	smoothne	compactn	concavity_	concave p	symmetry_	fractal_dimension_worst
8.589	153.4	0.0064	0.04904	0.05373	0.01587	0.03003	0.00619	25.38	17.33	184.6	2019	0.1622	0.6656	0.7119	0.2654	0.4601	0.1189
3.398	74.08	0.00523	0.01308	0.0186	0.0134	0.01389	0.00353	24.99	23.41	158.8	1956	0.1238	0.1866	0.2416	0.186	0.275	0.08902
4.585	94.03	0.00615	0.04006	0.03832	0.02058	0.0225	0.00457	23.57	25.53	152.5	1709	0.1444	0.4245	0.4504	0.243	0.3613	0.08758
3.445	27.23	0.00911	0.07458	0.05661	0.01867	0.05963	0.00921	14.91	26.5	98.87	567.7	0.2098	0.8663	0.6869	0.2575	0.6638	0.173
5.438	94.44	0.01149	0.02461	0.05688	0.01885	0.01756	0.00512	22.54	16.67	152.2	1575	0.1374	0.205	0.4	0.1625	0.2364	0.07678
2.217	27.19	0.00751	0.03345	0.03672	0.01137	0.02165	0.00508	15.47	23.75	103.4	741.6	0.1791	0.5249	0.5355	0.1741	0.3985	0.1244
3.18	53.91	0.00431	0.01382	0.02254	0.01039	0.01369	0.00218	22.88	27.66	153.2	1606	0.1442	0.2576	0.3784	0.1932	0.3063	0.08368
3.856	50.96	0.00881	0.03029	0.02488	0.01448	0.01486	0.00541	17.06	28.14	110.6	897	0.1654	0.3682	0.2678	0.1556	0.3196	0.1151
2.406	24.32	0.00573	0.03502	0.03553	0.01226	0.02143	0.00375	15.49	30.73	106.2	739.3	0.1703	0.5401	0.539	0.206	0.4378	0.1072
2.039	23.94	0.00715	0.07217	0.07743	0.01432	0.01789	0.01008	15.09	40.68	97.65	711.4	0.1853	1.058	1.105	0.221	0.4366	0.2075
2.466	40.51	0.00403	0.00927	0.01101	0.00759	0.0146	0.00304	19.19	33.88	123.8	1150	0.1181	0.1551	0.1459	0.09975	0.2948	0.08452
3.564	54.16	0.00577	0.04061	0.02791	0.01282	0.02008	0.00414	20.42	27.28	136.5	1299	0.1396	0.5609	0.3965	0.181	0.3792	0.1048
11.07	116.2	0.00314	0.08297	0.0889	0.0409	0.04484	0.01284	20.96	29.94	151.7	1332	0.1037	0.3903	0.3639	0.1767	0.3176	0.1023
2.903	36.58	0.00977	0.03126	0.05051	0.01992	0.02981	0.003	16.84	27.66	112	876.5	0.1131	0.1924	0.2322	0.1119	0.2809	0.06287
2.061	19.21	0.00643	0.05936	0.05501	0.01628	0.01961	0.00809	15.03	32.01	108.8	697.7	0.1651	0.7725	0.6943	0.2208	0.3596	0.1431
2.879	32.55	0.00561	0.0424	0.04741	0.0109	0.01857	0.00547	17.46	37.13	124.1	943.2	0.1678	0.6577	0.7026	0.1712	0.4218	0.1341
3.195	45.4	0.00572	0.01162	0.01998	0.01109	0.0141	0.00209	19.07	30.88	123.4	1138	0.1464	0.1871	0.2914	0.1609	0.3029	0.08216
3.854	54.18	0.00703	0.02501	0.03188	0.01297	0.01689	0.00414	20.96	31.48	136.8	1315	0.1789	0.4233	0.4784	0.2073	0.3706	0.1142
5.865	112.4	0.00649	0.01893	0.03391	0.01521	0.01356	0.002	27.32	30.88	186.8	2398	0.1512	0.315	0.5372	0.2388	0.2768	0.07615
2.058	23.56	0.00846	0.0146	0.02387	0.01315	0.0198	0.0023	15.11	19.26	99.7	711.2	0.144	0.1773	0.239	0.1288	0.2977	0.07259
1.383	14.67	0.0041	0.01898	0.01698	0.00649	0.01678	0.00243	14.5	20.49	96.09	630.5	0.1312	0.2776	0.189	0.07283	0.3184	0.08183
1.909	15.7	0.00961	0.01432	0.01985	0.01421	0.02027	0.00297	10.23	15.66	65.13	314.9	0.1324	0.1148	0.08867	0.06227	0.245	0.07773
3.384	44.91	0.00679	0.05328	0.06446	0.02252	0.03672	0.00439	18.07	19.08	125.1	980.9	0.139	0.5954	0.6305	0.2393	0.4667	0.09946
4.303	93.99	0.00473	0.01259	0.01715	0.01038	0.01083	0.00199	29.17	35.59	188	2615	0.1401	0.26	0.3155	0.2009	0.2822	0.07526
5.455	102.6	0.00605	0.01882	0.02741	0.0113	0.01468	0.0028	26.46	31.56	177	2215	0.1805	0.3578	0.4695	0.2095	0.3613	0.09564
7.276	111.4	0.00803	0.03799	0.03732	0.02397	0.02308	0.00744	22.25	21.4	152.4	1461	0.1545	0.3949	0.3853	0.255	0.4066	0.1059
2.11	21.05	0.00445	0.03055	0.02681	0.01352	0.01454	0.00371	17.62	33.21	122.4	896.9	0.1525	0.6643	0.5539	0.2701	0.4264	0.1275
5.632	93.54	0.01075	0.02722	0.05081	0.01911	0.02293	0.00422	21.31	27.26	139.9	1403	0.1338	0.2117	0.3446	0.149	0.2341	0.07421
3.498	43.5	0.00523	0.03057	0.03576	0.01083	0.01768	0.00297	20.27	36.71	149.3	1269	0.1641	0.611	0.6335	0.2024	0.4027	0.09876
4.655	61.1	0.00563	0.03033	0.03407	0.01354	0.01925	0.00374	20.01	19.52	134.9	1227	0.1255	0.2812	0.2489	0.1456	0.2756	0.07919

Figure 6:Dataset example 2

2.2. Review and Analysis of existing work in problem domain

2.2.1. Breast Cancer Risk Prediction using XGBoost and Random Forest Algorithm

Authors: Sajib Kabiraj, M.Raihan, Nasif Alvi, Marina Afrin, Laboni Akter, Shwami Akhter Sohagi, Etu Podder

Date of Conference: 01-03 July 2020

Publisher: IEEE (Institute of Electrical and Electronics engineering)

INSPEC Accession Number: 20064190

The research paper "Breast cancer Risk Prediction using XGBoost and Random Forest Algorithm" explores the use of Random Forest and Extreme Gradient Boosting algorithms to predict breast cancer. The study uses a dataset of 275 instances with 12 features and achieves an accuracy of 74.73% for Random Forest and 73.63% for XGBoost. The study's methodology includes data collection, pre-processing, training, and application of algorithms. The authors plan to increase the sample size of the dataset in future research to improve the reliability, accuracy, and effectiveness of their analysis.

2.2.2. A Precise Breast Cancer Detection Approach Using Ensemble of Random Forest with AdaBoost

Authors: Tanbin Islam Rohan, Awan-ur-Rahaman, Abu Bakar Siddik, Monira Islam, Md. Salah Uddin Yusuf

Date of Conference: 11-12 July 2019

Publisher: IEEE (Institute of Electrical and Electronics engineering)

INSPEC Accession Number: 19470516

Breast cancer is a significant health concern, leading to numerous death each year. Early detection is key to curing the diseases. Therefore, a reliable and accurate system for classifying benign and malignant tumour types of breast cancer is crucial. This paper presents a supervised machine learning model for this classification task using the

Wisconsin Breast Cancer dataset from the UCI machine learning repository. The dataset comprises 458 benign instances and 241 malignant instances, totalling 699 instances, 11 features, and 10 attributes.

The Random Forest (RF) ensemble learning method is implemented with the AdaBoost algorithm, which improved the performance metrics in binary classification between tumour classes. To enhance the estimation of model prediction performance, 10 -fold-cross-validation is applied. The proposed structure achieved an accuracy of 98.5714% in the testing phase, along with sensitivity and specificity of 100% and 96.296% respectively. The Matthews Correlation Coefficient was calculated as 0.97, validating the structure as a pure binary classifier for this work. The proposed structure outperformed the conventional RF classifier in classifying tumour types and also enhanced the performance of conventional classifiers.

2.2.3. Using Random Forest Algorithm for Breast Cancer Diagnosis

Authors: Bin Dai, Rung-Ching Chen, Shun-Zhi Zhu, Wei-Wei Zhang

Date of Conference: 06-08 December 2018

Publisher: IEEE (Institute of Electrical and Electronics engineering)

INSPEC Accession Number: 18472865

This paper discusses the use of the Random Forest algorithm to analyze the diagnosis of breast cancer cases. The Random Forest algorithm can combine the characteristics of multiple eigenvalues, and the combined results of multiple decision trees can enhance the prediction accuracy. By using the ensemble learning method of random forests, the results of multiple weak classifiers can be combined to produce accurate classification results.

The Random Forest algorithm is used in this paper to discuss the case of breast cancer case diagnosis and achieve high prediction accuracy. This has practical significance for auxiliary medical diagnosis. The use of machine learning for medical diagnosis and the accuracy of these diagnoses is a major change and an inevitable future direction for medical models

2.2.4. Performance Comparison of Different Machine Learning Techniques for Early Prediction of Breast Cancer using Wisconsin Breast Cancer Dataset.

Authors: Atajan Rovshenov, Serhat Peker

Date of Conference: 15-16 December 2022

Publisher: IEEE (Institute of Electrical and Electronics engineering)

Cancer, a significant global health concern, is increasingly prevalent and a leading cause of mortality. Among various types, breast cancer is particularly widespread among women. Early detection is crucial for improving survival rates and reducing treatment costs. However, current early diagnosis methods face challenges such as the need for extensive human resources, long-term effects, and limited accessibility.

Addressing these issues requires user-friendly technologies that provide reliable results comparable to traditional methods and are accessible to everyone. Artificial Intelligence (AI) techniques offer a promising avenue for early breast cancer diagnosis. This study focuses on classifying features in breast cancer images as either benign or malignant. Utilizing Artificial Neural Network, Support Vector Machine, and Random Forest algorithms, the research employed the Wisconsin Breast Cancer dataset for experiments.

The experimental results indicate that the Artificial Neural Network algorithm achieved the highest success rate at 99%. This suggests that the proposed classification technique can effectively identify breast cancer in its early stages. These findings are anticipated to inspire further research into innovative approaches for early breast cancer detection.

2.2.5. Machine Learning Approach for Breast Cancer Prediction: A Review

Authors: Yash Wankhade, Shrividya Toutam, Khushboo Thakre, Kamlesh Kalbande, Prasheel Thakre

Date of Conference: 04-06 May 2023

Publisher: IEEE (Institute of Electrical and Electronics engineering)

Breast cancer is a complex and widespread illness impacting millions of women globally. Timely and accurate diagnosis plays a crucial role in effective therapy and improved patient outcomes. In recent years, there has been a growing interest in developing predictive models and machine learning algorithms for the detection and diagnosis of breast cancer. This research study aims to provide a comprehensive overview of the latest breast cancer prognostic models, encompassing risk assessment, diagnosis, and prognosis.

The paper explores various data types, such as clinical, genetic, and imaging data, employed in breast cancer prediction. Additionally, it delves into the diverse machine learning techniques utilized, including Support Vector Machines (SVM), naïve Bayes, and random forests. The research study includes a comparative analysis of different algorithms along with their methodologies, offering insights into their respective strengths and weaknesses in the context of breast cancer prediction.

2.2.6. Summarised Review and Analysis

The reviewed papers collectively underscore the pressing need for advancements in breast cancer detection and diagnosis. Recognizing breast cancer as a complex and widespread health concern, the studies emphasize the significance of correct diagnosis and early detection for effective treatment and improved patient outcomes.

One prominent trend in recent research involves the development of predictive models and machine learning algorithms for breast cancer detection. The studies highlight the diverse data types employed in these models, ranging from clinical and genetic information to imaging data. This holistic approach reflects the multidimensional nature of breast cancer, acknowledging the importance of considering various aspects for accurate predictions.

The machine learning techniques utilized across these papers encompass Support Vector Machines (SVM), naïve Bayes, and random forests. The inclusion of a comparative analysis provides valuable insights into the strengths and weaknesses of each algorithm. Notably, the studies showcase the potential of Artificial Neural Network algorithms, with one study achieving a remarkable 99% success rate in early breast cancer detection.

Overall, these papers contribute to the ongoing efforts in breast cancer research by presenting a thorough overview of the latest prognostic models. By addressing different data types and employing various machine learning techniques, these studies pave the way for more effective and nuanced approaches to breast cancer prediction, diagnosis, and prognosis. The collective findings encourage further exploration and innovation in the quest for improved early detection and treatment strategies.

3. Solution

3.1. Algorithms

Some of the commonly known examples of machine learning algorithms are listed as below:

- **Linear Regression:** This is a simple yet powerful algorithm used for predicting continuous values. It establishes a linear relationship between dependent and independent variables and uses the best fit line to predict the output. For example, it can be used to predict house prices based on factors like size, location, and age. (Brownlee, 2023)

- **Decision Trees:** Decision trees are flowchart-like structures used for decision making. They split the population or sample into subsets based on attribute values. For instance, a decision tree could be used to determine whether a person has diabetes based on symptoms and medical history. (IBM, 2024)

- **Random Forest:** Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputting the class that is the mode of the classes of the individual trees. It can handle both numerical and categorical data and is particularly useful for handling large datasets. (IBM, 2024)

- **K-Nearest Neighbours (KNN):** KNN is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. It stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). (IBM, 2024)

- **Support Vector Machines (SVM):** SVM is a supervised machine learning algorithm used for classification and regression analysis. It maps the input data into high-dimensional feature spaces and finds a hyperplane that maximally separates the classes. (aslanyan, 2023)

- **Naive Bayes:** This is a probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions between the features. Naive Bayes classifiers are highly scalable and easy to build. They are often used in text classification and spam filtering applications. (IBM, 2024)

- **Neural Networks:** Neural networks are a series of algorithms that are designed to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. They are used in image recognition, natural language processing, and speech recognition. (IBM, 2024)

3.2. Elaboration of the considered methodologies

3.2.1. K-Nearest Neighbour Algorithm

The K-Nearest Neighbours (KNN) algorithm is a popular machine learning technique used for both classification and regression tasks. It operates on the principle that similar data points tend to have similar labels or values. During the training phase, the KNN algorithm stores the entire training dataset as a reference. When making predictions, it calculates the distance between the input data point and all the training examples, using a chosen distance metric such as Euclidean distance. (Srivastava, 2024)

Next, the algorithm identifies the K nearest neighbours to the input data point based on their distances. In the case of classification, the algorithm assigns the most common class label among the K neighbours as the predicted label for the input data point. For regression, it calculates the average or weighted average of the target values of the K neighbours to predict the value for the input data point. (Srivastava, 2024)

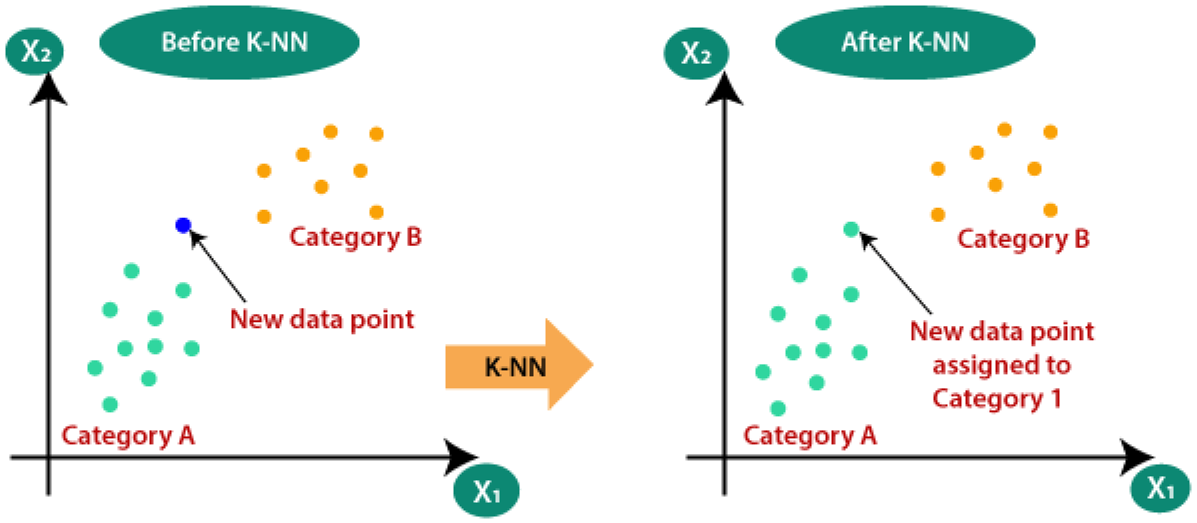


Figure 7:KNN Algorithm

The choice of K is crucial in the KNN algorithm. Too small a value of K can lead to unstable decision boundaries, while too large a value can lead to overfitting. The optimal value of K can be determined using methods such as cross-validation, where the error rate is plotted against different K values, and the K value that gives the smallest error rate is chosen. (IBM, 2024)

Despite its simplicity, the KNN algorithm has limitations. It's sensitive to the choice of distance metric and the value of K, and it can struggle with high-dimensional data. Additionally, because it stores all training data, it can be computationally expensive and memory-intensive for large datasets. (IBM, 2024)

However, KNN is widely used in various fields due to its simplicity and effectiveness. It's used in areas like disease prediction, handwriting recognition, image classification, and financial market predictions, among others. (Srivastava, 2024)

3.2.2. Random Forest Algorithm

The Random Forest algorithm is a supervised learning method that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes of the individual trees. It is a powerful machine learning algorithm that uses ensemble learning methods to create a strong predictive model. (IBM, 2024)

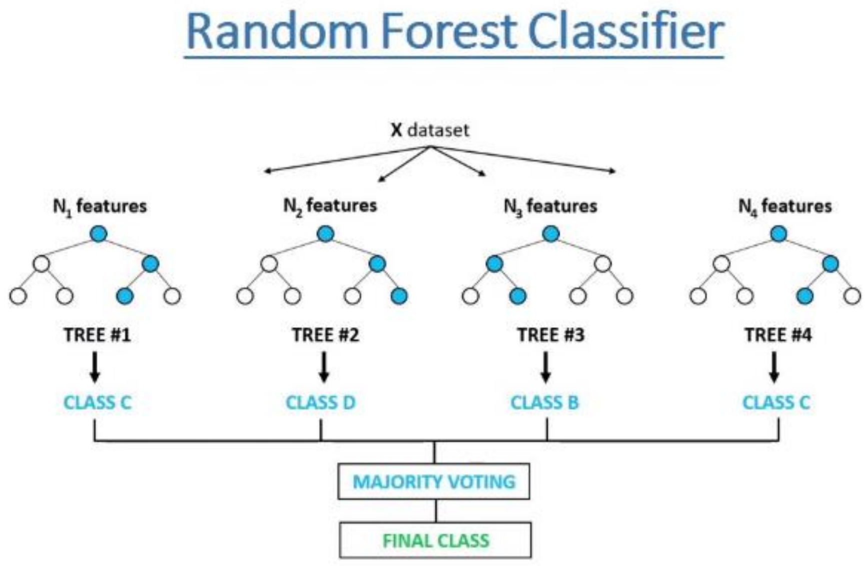


Figure 8:Random Forest (David, 2022)

The "forest" it builds is an ensemble of decision trees, usually trained with the bagging method. Bagging, also known as bootstrap aggregation, is a technique for reducing variance within a noisy dataset. In this method, a random sample of data in a training set is selected with replacement—meaning that the individual data points can be chosen more than once. After several data samples are generated, these models are then trained independently. Depending on the type of task—i.e., regression or classification—the average or majority of those predictions yield a more accurate estimate. One of the key benefits of Random Forest is its ability to determine feature importance. This is done by evaluating the contribution of each variable to the model. There are several ways to measure feature importance, such as Gini importance, mean decrease in impurity (MDI), and permutation importance, also known as mean decrease accuracy (MDA). These measures assess how much the model’s accuracy decreases when a given variable is excluded. However, Random Forest also has certain challenges. One of them is that it might be biased in favour of attributes with more levels when dealing with categorical variables with different numbers of levels. Additionally, if the data contains groups of correlated features of similar relevance for the output, then smaller groups are favoured over larger groups. Also, the permutation procedure may fail to identify important features when there are collinear features. (Donges, 2023)

Despite these challenges, Random Forest has found wide application in various fields like finance, healthcare, and e-commerce. It can be used for tasks like evaluating customers with high credit risk, detecting fraud, option pricing problems, gene expression classification, biomarker discovery, and sequence annotation, and recommendation engines for cross-sell purposes. (IBM, 2024)

3.2.3. SVM algorithm

The Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression tasks. However, it is best suited for classification problems. The primary goal of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that separates data points belonging to different classes in the feature space. The hyperplane tries to maximize the margin between the closest points of different classes. (Saini, 2024)

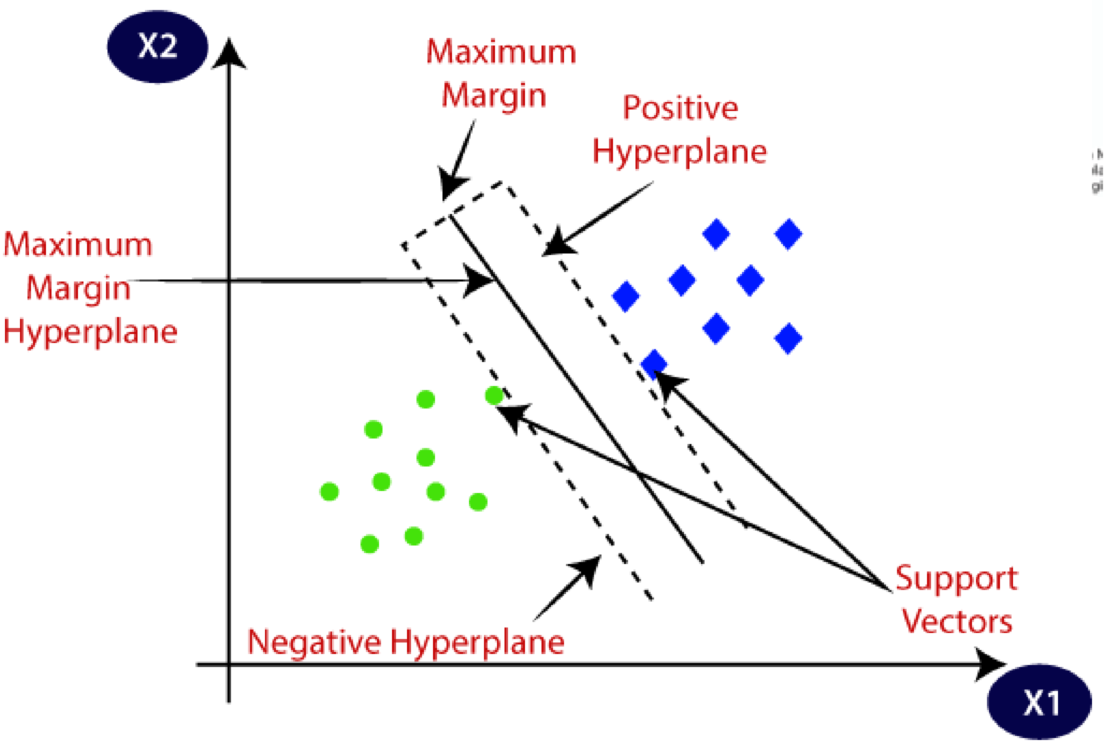


Figure 9:Svm algorithm

The SVM algorithm works by defining the model in terms of the support vectors only, i.e., the data points that are closest to the hyperplane. This allows the algorithm to enjoy some natural speed-ups. The best hyperplane is the one that has the maximum distance from both classes, and this is achieved by finding different hyperplanes which classify the labels in the best way. Then it chooses the one which is farthest from the data points or the one which has a maximum margin. (Saini, 2024)

One of the key features of SVM is its ability to work with a non-linear dataset using the "Kernel Trick". This involves transforming the input data into a higher-dimensional space where it can be linearly separated. Various kernel functions can be used for this transformation, and the choice of kernel function is often a critical aspect of the SVM algorithm. SVMs are effective in high-dimensional cases and are memory efficient as they use a subset of training points in the decision function called support vectors. Different kernel functions can be specified for the decision functions and it's possible to specify custom kernels. (Saini, 2024)

SVMs have also been extended to multi-class classification problems through methods like one-vs-all or one-vs-one strategies. In the one-vs-all approach, a separate SVM is trained for each class, and the classifier with the highest-output function assigns the class. For the one-vs-one approach, every classifier assigns the instance to one of the two classes, then the vote for the assigned class is increased by one vote, and finally, the class with the most votes determines the instance classification. (Saini, 2024)

3.3. Pseudocode

```
START

IMPORT libraries

{

    IMPORT Dataset

    DISPLAY dataset as a data frame

    CLEAN Dataset

    DISPLAY generated charts

    INITIATE X and y Sets

    SPLIT Dataset to training SET and test SET

    Fit Training_features and classes INTO KNNClassifier()

    CALCULATE Accuracy and metrics

    DISPLAY Accuracy and metrics

    Fit Training_features and classes INTO SVM()

    CALCULATE Accuracy and metrics

    DISPLAY Accuracy and metrics

    Fit Training_features and classes INTO RandomForest()

    CALCULATE Accuracy metrics

    DISPLAY Accuracy metrics
    PERFORM Data tuning on RandomForest()

    DISPLAY Best hyperparameters for model fitting

    DISPLAY  performance metrics

    DISPLAY Confusion Matrix

}

STOP
```

3.4. Diagrammatic representation of the system

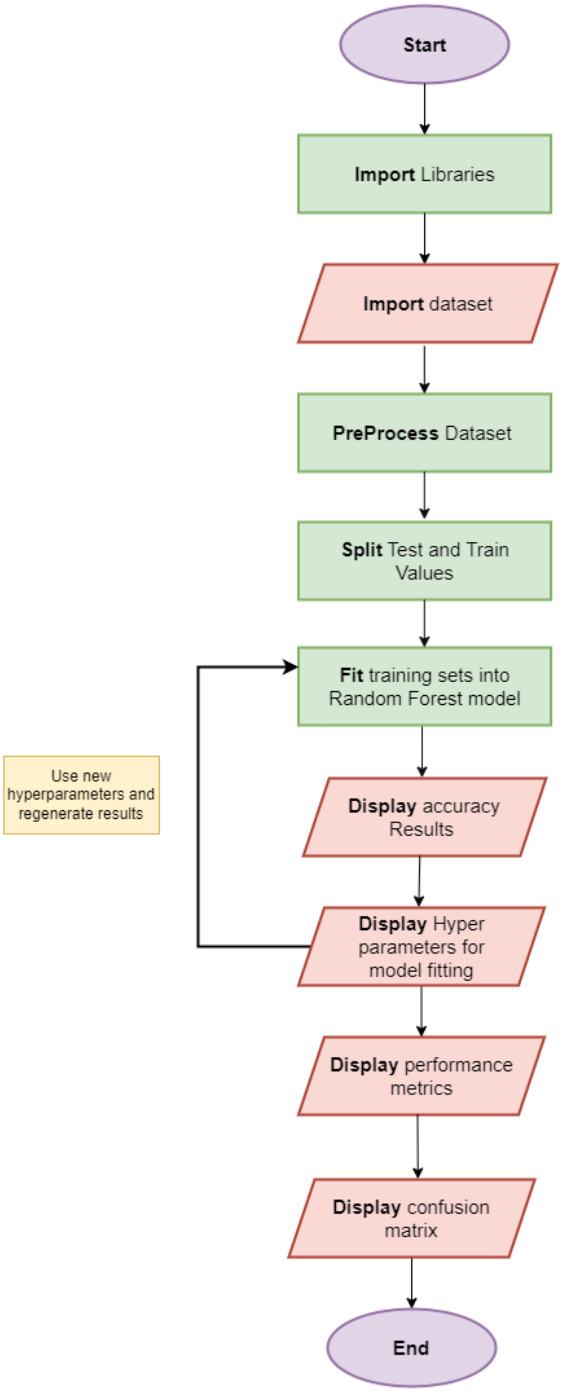

Start
Import Libraries
Import dataset
PreProcess Dataset
Split Test and Train Values
Fit training sets into Random Forest model
Use new hyperparameters and regenerate results
Display accuracy Results
Display Hyper parameters for model fitting
Display performance metrics
Display confusion matrix
End

Figure 10:Flowchart for the system

3.5. Development Process

3.5.1. Tools Used

The tools used in the project are listed below:

- **Jupyter Notebook:** Jupyter Notebook is an open-source web application used for creating and sharing documents that contain live code, equations, visualizations, and narrative text. It is a powerful tool for interactively developing and presenting data science projects. A Jupyter Notebook integrates code and its output into a single document that combines visualizations, narrative text, mathematical equations, and other rich media. This makes the work more transparent, understandable, repeatable, and shareable. It is widely used in the data science workflow at companies across the globe. As part of the open-source Project Jupyter, Jupyter Notebooks are completely free. Although it is possible to use many different programming languages in Jupyter Notebooks, Python is the most common use case. Jupyter Notebooks are primarily used by data professionals, particularly data analysts and data scientists. (Driscoll, 2023)

3.5.2. Toolkits used

- **NumPy:** NumPy was employed due to its efficiency in handling and manipulating arrays. Its seamless integration with other libraries like OpenCV and Matplotlib made it a preferred choice. NumPy primarily played a role in storing values within arrays.

- **Pandas:** Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.

- **Matplotlib:** Matplotlib, the plotting library, was combined with NumPy and to visually represent the confusion matrix. This integration facilitated a more understandable visualization of the outcomes of leaf image processing.

- **Sci-kit Learn:** The sci-kit learn library was instrumental in implementing the Random Forest algorithm. It played a role in splitting the dataset into training and testing sets. Additionally, Sci-kit Learn was employed for model evaluation, utilizing metrics such as accuracy, precision, and recall. The library also proved useful in hyperparameter tuning, employing tools like RandomizedSearchCV to fine-tune the model's parameters.

- **RandomForestClassifier**: A classifier based on the concept of decision trees. Each tree gives a classification, and the final prediction is obtained by averaging the classifications of individual trees.

- **KNeighborsClassifier:** A classifier implementing the k-nearest neighbors vote.

- **StandardScaler:** Standardize features by removing the mean and scaling to unit variance.

- **Metrics:** Collection of scoring metric functions.

- **ConfusionMatrix:** Confusion matrix object.

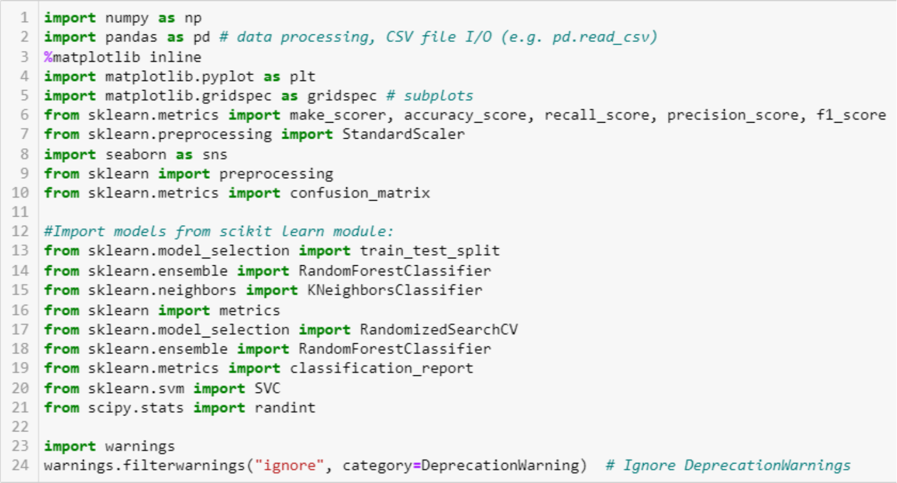
3.6. Explanation of the development process

3.6.1. Brief description of the development process

- **Importing Libraries and data loading:** he initial phase of any data science endeavour involves importing the required libraries and loading the data. The necessary libraries and data are thus imported, and the data is loaded into a pandas dataframe.
- **Data Understanding:** Upon loading the data, it's crucial to comprehend its contents. This entails examining the data's shape, variable types, and understanding the distribution of these variables.
- **Data Manipulation:** This stage encompasses cleaning the data and readying it for analysis by addressing missing values, transforming categorical variables into numerical ones, normalizing numerical variables, etc.
- **Data visualization:** Following the data cleaning process, visualizations are generated to enhance our understanding of the data. These may include histograms, boxplots, scatter plots, etc.
- **Data Pre-processing:** The data is converted into a format that can be easily interpreted by the machine learning algorithm. The data is preprocessed to ready it for model training. This includes feature scaling, managing categorical variables, and dividing the dataset into training and testing sets.
- **ML Model Evaluation and Prediction:** Machine learning models are selected and trained on the pre-processed data. The models used for churn prediction are Random Forests, Support Vector Machines (SVM), and KNN classifier. The models are evaluated using metrics like accuracy, precision, recall, and F1- score. Hyperparameter tuning is performed using RandomizedSearchCV to optimize model performance. The final model is then used for predicting a cancerous tumour.

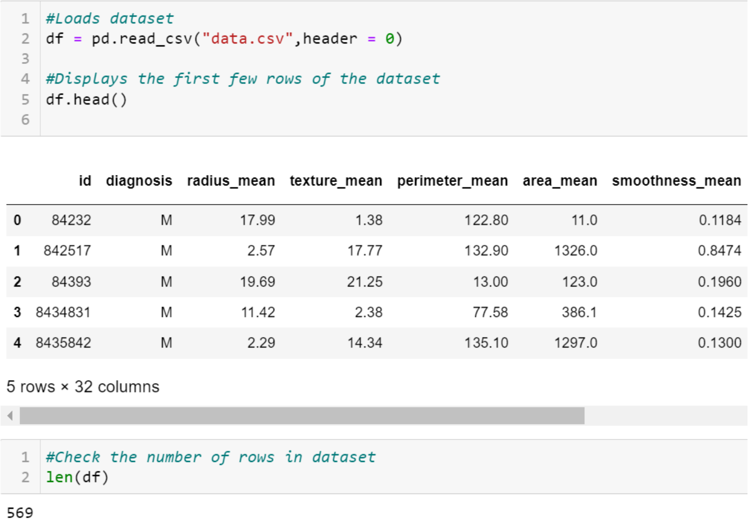
3.6.2. Importing Required Libraries

This script imports several libraries for data processing, machine learning, and data visualization. Numpy and pandas are used for handling and analyzing data. Matplotlib and seaborn are used for creating plots and visualizations. Scikit-learn is used for machine learning tasks such as splitting data into training and testing sets, creating classifiers, and evaluating their performance. The script also uses scipy for statistical functions and ignores deprecation warnings.



```python
import numpy as np
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
%matplotlib inline
import matplotlib.pyplot as plt
import matplotlib.gridspec as gridspec # subplots
from sklearn.metrics import make_scorer, accuracy_score, recall_score, precision_score, f1_score
from sklearn.preprocessing import StandardScaler
import seaborn as sns
from sklearn import preprocessing
from sklearn.metrics import confusion_matrix

#Import models from scikit learn module:
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
from sklearn.model_selection import RandomizedSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report
from sklearn.svm import SVC
from scipy.stats import randint

import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)  # Ignore DeprecationWarnings
```

Figure 11:Importing necessary libraries



```python
#Loads dataset
df = pd.read_csv("data.csv",header = 0)

#Displays the first few rows of the dataset
df.head()

```

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean
0	84232	M	17.99	1.38	122.80	11.0	0.1184
1	842517	M	2.57	17.77	132.90	1326.0	0.8474
2	84393	M	19.69	21.25	13.00	123.0	0.1960
3	8434831	M	11.42	2.38	77.58	386.1	0.1425
4	8435842	M	2.29	14.34	135.10	1297.0	0.1300

5 rows × 32 columns

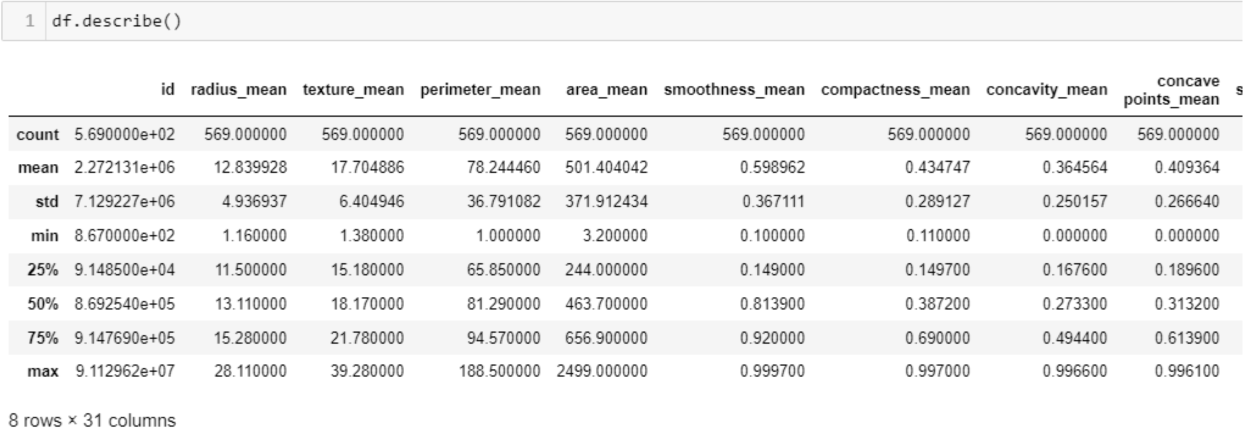
```python
#Check the number of rows in dataset
len(df)
```

```
569
```

Figure 12:Loading Dataset

3.6.3. Data Understanding

The process of comprehending the data involves examining its shape, variable types, and variable distributions.



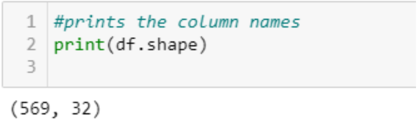
```
df.describe()
```

	id	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	s
count	5.690000e+02	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	569.000000	
mean	2.272131e+06	12.839928	17.704886	78.244460	501.404042	0.598962	0.434747	0.364564	0.409364	
std	7.129227e+06	4.936937	6.404946	36.791082	371.912434	0.367111	0.289127	0.250157	0.266640	
min	8.670000e+02	1.160000	1.380000	1.000000	3.200000	0.100000	0.110000	0.000000	0.000000	
25%	9.148500e+04	11.500000	15.180000	65.850000	244.000000	0.149000	0.149700	0.167600	0.189600	
50%	8.692540e+05	13.110000	18.170000	81.290000	463.700000	0.813900	0.387200	0.273300	0.313200	
75%	9.147690e+05	15.280000	21.780000	94.570000	656.900000	0.920000	0.690000	0.494400	0.613900	
max	9.112962e+07	28.110000	39.280000	188.500000	2499.000000	0.999700	0.997000	0.996600	0.996100	

8 rows × 31 columns

Figure 13: Dataset Stats

Here, in the above figure, the. describe () is used to generate a descriptive statistic of the dataframe. It summarizes and organises a set of data points as shown above.

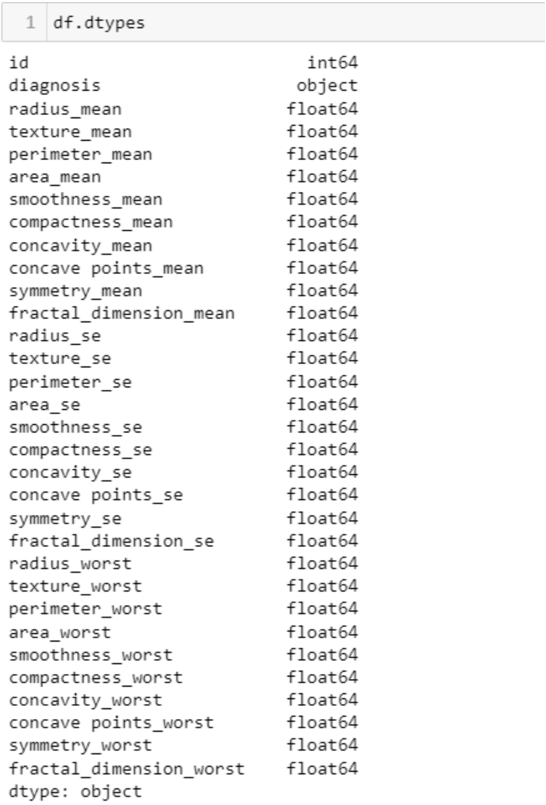


```
#prints the column names
print(df.shape)

```

```
(569, 32)
```

Figure 14:Dataframe shape

Here, in the figure above, the .shape() is used to get the dimension of the dataframe. The first element is the number of rows and the second element is the number of columns.



```
df.dtypes
```

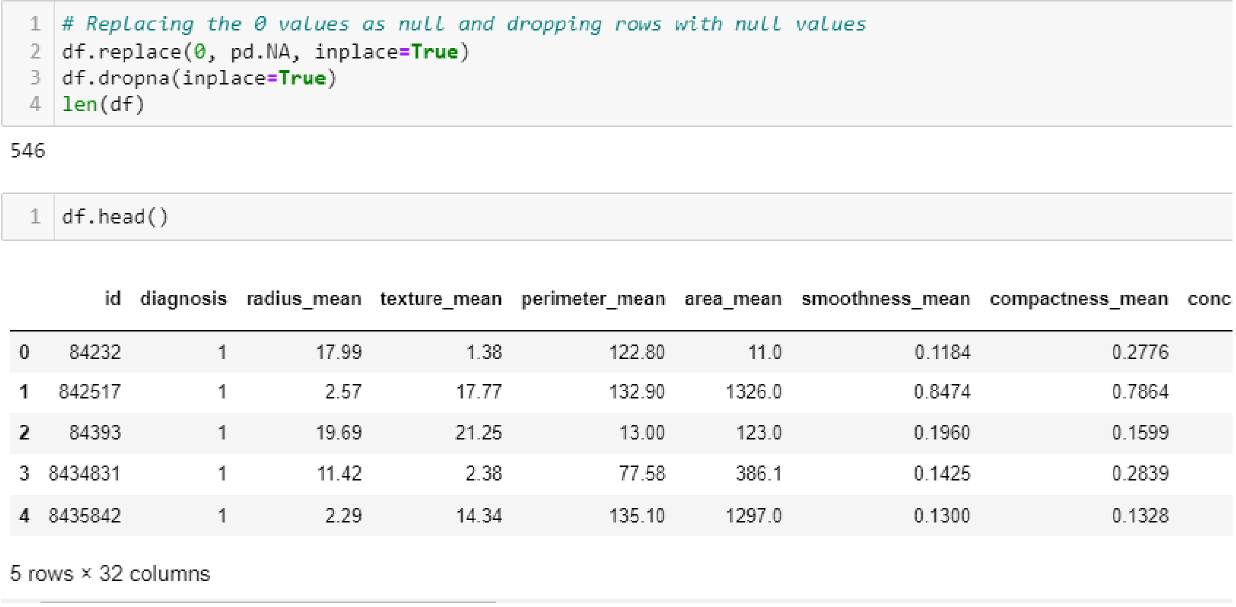
```
id                          int64
diagnosis                  object
radius_mean               float64
texture_mean              float64
perimeter_mean            float64
area_mean                 float64
smoothness_mean           float64
compactness_mean          float64
concavity_mean            float64
concave points_mean       float64
symmetry_mean             float64
fractal_dimension_mean    float64
radius_se                 float64
texture_se                float64
perimeter_se              float64
area_se                   float64
smoothness_se             float64
compactness_se            float64
concavity_se              float64
concave points_se         float64
symmetry_se               float64
fractal_dimension_se      float64
radius_worst              float64
texture_worst             float64
perimeter_worst           float64
area_worst                float64
smoothness_worst          float64
compactness_worst         float64
concavity_worst           float64
concave points_worst      float64
symmetry_worst            float64
fractal_dimension_worst   float64
dtype: object
```

Figure 15:Datatypes of the columns

In the above figure, the .dtypes method has been employed to retrieve the data types of all the columns within the dataframe. The three figures presented above illustrate the approach taken during the initial phase of model development for understanding the data

3.6.4. Data Manipulation

Data manipulation is a crucial stage in both data analysis and machine learning workflows. This process encompasses tasks such as addressing missing values, transforming data, creating new features, reducing dimensionality, normalizing numerical data, and integrating information from various sources. Each of these steps plays a vital role in preparing the data for subsequent analysis or model development, ultimately enhancing the performance and accuracy of the models.



```python
# Replacing the 0 values as null and dropping rows with null values
df.replace(0, pd.NA, inplace=True)
df.dropna(inplace=True)
len(df)
```

```
546
```

```python
df.head()
```

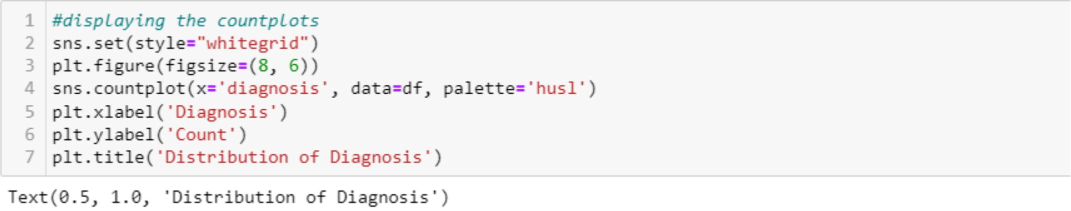
	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	conc
0	84232	1	17.99	1.38	122.80	11.0	0.1184	0.2776	
1	842517	1	2.57	17.77	132.90	1326.0	0.8474	0.7864	
2	84393	1	19.69	21.25	13.00	123.0	0.1960	0.1599	
3	8434831	1	11.42	2.38	77.58	386.1	0.1425	0.2839	
4	8435842	1	2.29	14.34	135.10	1297.0	0.1300	0.1328	

5 rows × 32 columns

Figure 16:Data Manipulation

3.6.5. Data Visualisation

Data visualization involves presenting information and data in a graphical format, utilizing visual elements such as charts, graphs, and maps to offer an accessible means of observing and comprehending trends, outliers, and patterns within the data. The following are some of the data visualizations conducted during the construction of the prediction model.



```python
#displaying the countplots
sns.set(style="whitegrid")
plt.figure(figsize=(8, 6))
sns.countplot(x='diagnosis', data=df, palette='husl')
plt.xlabel('Diagnosis')
plt.ylabel('Count')
plt.title('Distribution of Diagnosis')
```

```
Text(0.5, 1.0, 'Distribution of Diagnosis')
```

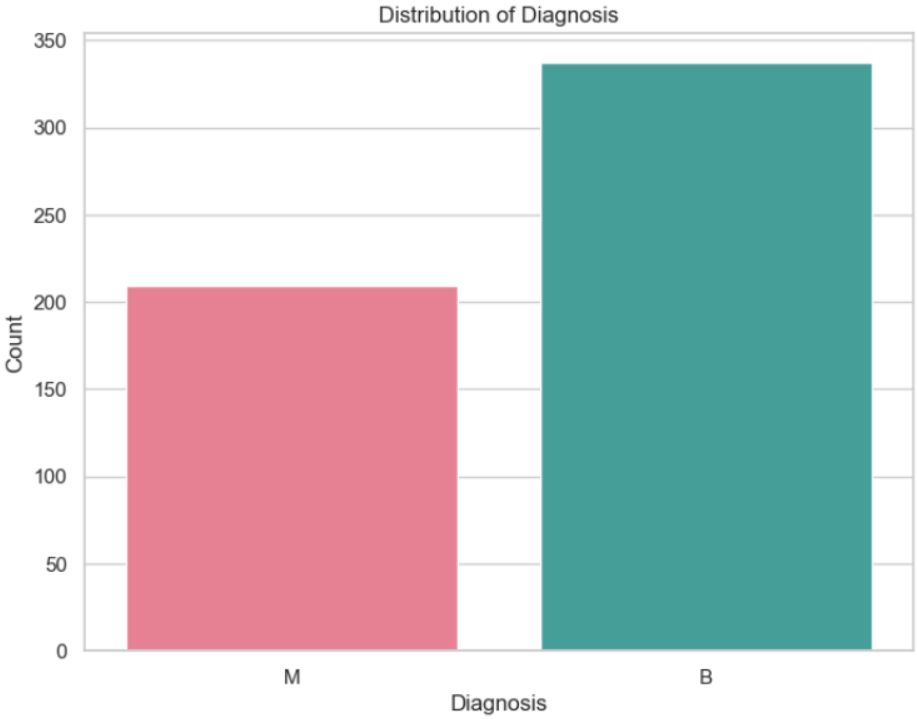
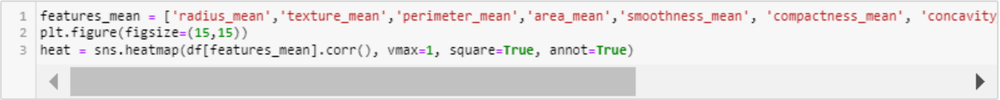
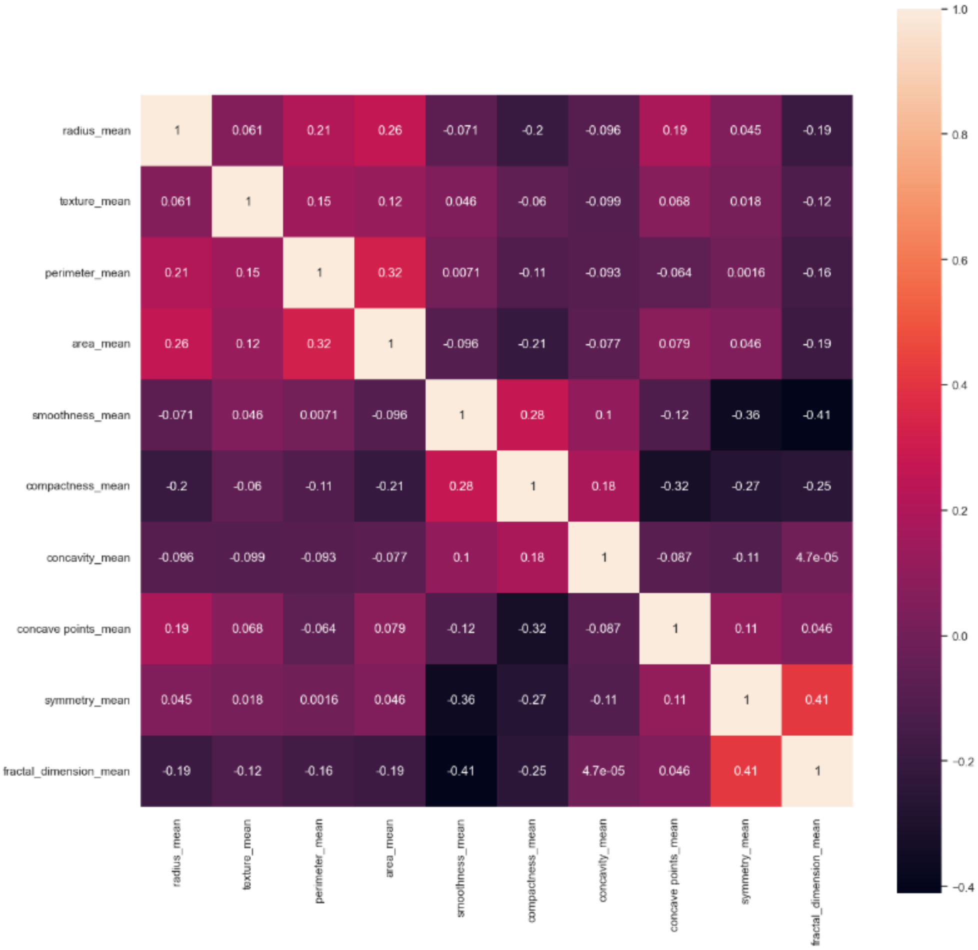


Figure 17:Count Plot

in the provided code, a count plot is used to visualize the distribution of the 'diagnosis' column. A count plot is specifically suitable for categorical data, such as the diagnosis of cases (malignant or benign in this context)



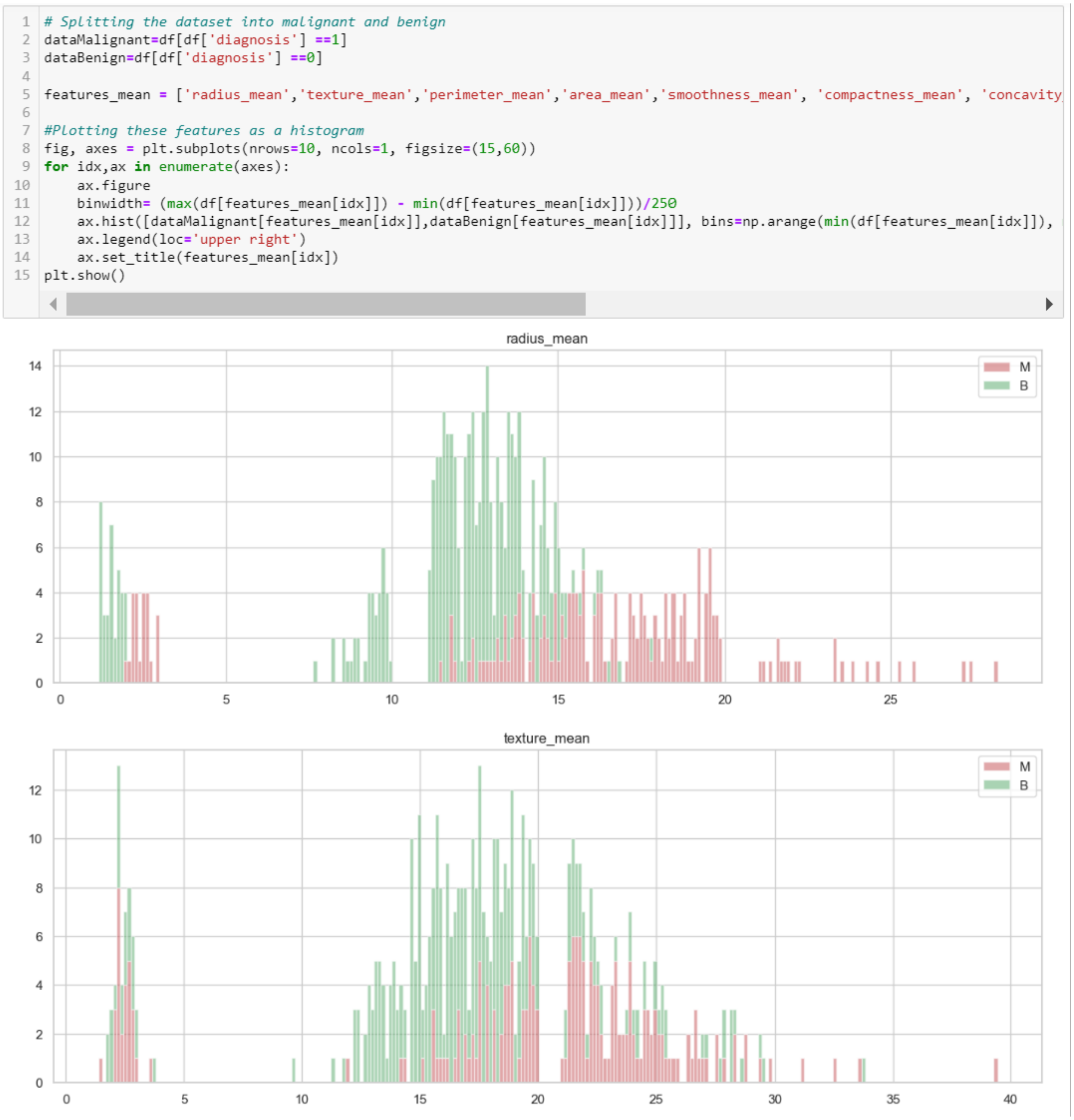
```
features_mean = ['radius_mean','texture_mean','perimeter_mean','area_mean','smoothness_mean', 'compactness_mean', 'concavity
plt.figure(figsize=(15,15))
heat = sns.heatmap(df[features_mean].corr(), vmax=1, square=True, annot=True)
```



	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimension_mean
radius_mean	1	0.061	0.21	0.26	-0.071	-0.2	-0.096	0.19	0.045	-0.19
texture_mean	0.061	1	0.15	0.12	0.046	-0.06	-0.099	0.068	0.018	-0.12
perimeter_mean	0.21	0.15	1	0.32	0.0071	-0.11	-0.093	-0.064	0.0016	-0.16
area_mean	0.26	0.12	0.32	1	-0.096	-0.21	-0.077	0.079	0.046	-0.19
smoothness_mean	-0.071	0.046	0.0071	-0.096	1	0.28	0.1	-0.12	-0.36	-0.41
compactness_mean	-0.2	-0.06	-0.11	-0.21	0.28	1	0.18	-0.32	-0.27	-0.25
concavity_mean	-0.096	-0.099	-0.093	-0.077	0.1	0.18	1	-0.087	-0.11	4.7e-05
concave points_mean	0.19	0.068	-0.064	0.079	-0.12	-0.32	-0.087	1	0.11	0.046
symmetry_mean	0.045	0.018	0.0016	0.046	-0.36	-0.27	-0.11	0.11	1	0.41
fractal_dimension_mean	-0.19	-0.12	-0.16	-0.19	-0.41	-0.25	4.7e-05	0.046	0.41	1

Figure 18:Heatmap

In the provided code, a heatmap is created using Seaborn to visualize the correlation matrix of a subset of features (features_mean) from the dataframe (df). This heatmap provides a visual representation of the pairwise correlations between the selected features. Positive correlations are indicated by warmer colours, negative correlations by cooler colours, and the intensity of the colour reflects the strength of the correlation. The numerical annotations provide the exact correlation coefficients.



```
# Splitting the dataset into malignant and benign
dataMalignant=df[df['diagnosis'] ==1]
dataBenign=df[df['diagnosis'] ==0]

features_mean = ['radius_mean','texture_mean','perimeter_mean','area_mean','smoothness_mean', 'compactness_mean', 'concavity

#Plotting these features as a histogram
fig, axes = plt.subplots(nrows=10, ncols=1, figsize=(15,60))
for idx,ax in enumerate(axes):
    ax.figure
    binwidth= (max(df[features_mean[idx]]) - min(df[features_mean[idx]]))/250
    ax.hist([dataMalignant[features_mean[idx]],dataBenign[features_mean[idx]]], bins=np.arange(min(df[features_mean[idx]]),
    ax.legend(loc='upper right')
    ax.set_title(features_mean[idx])
plt.show()
```

Figure 19:Histogram 1

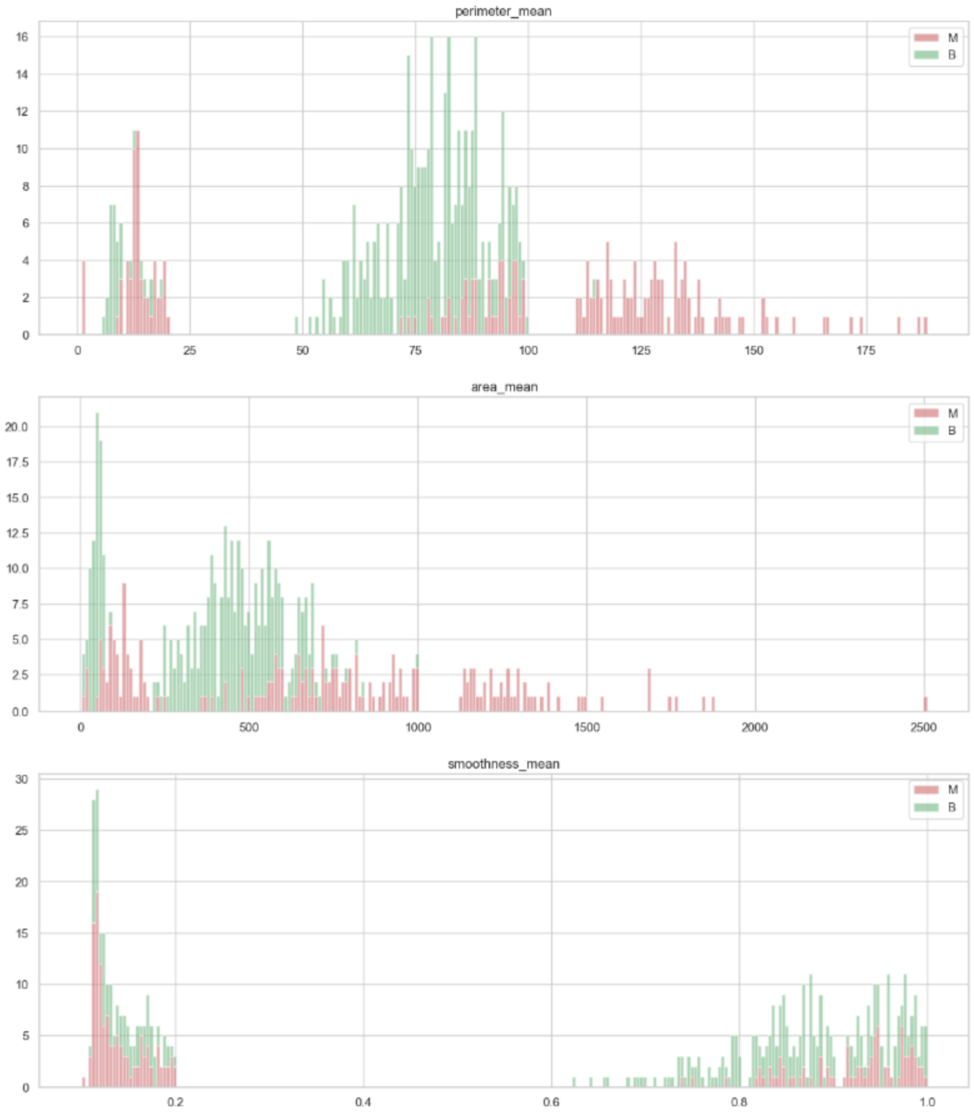


Figure 20:Histogram 2

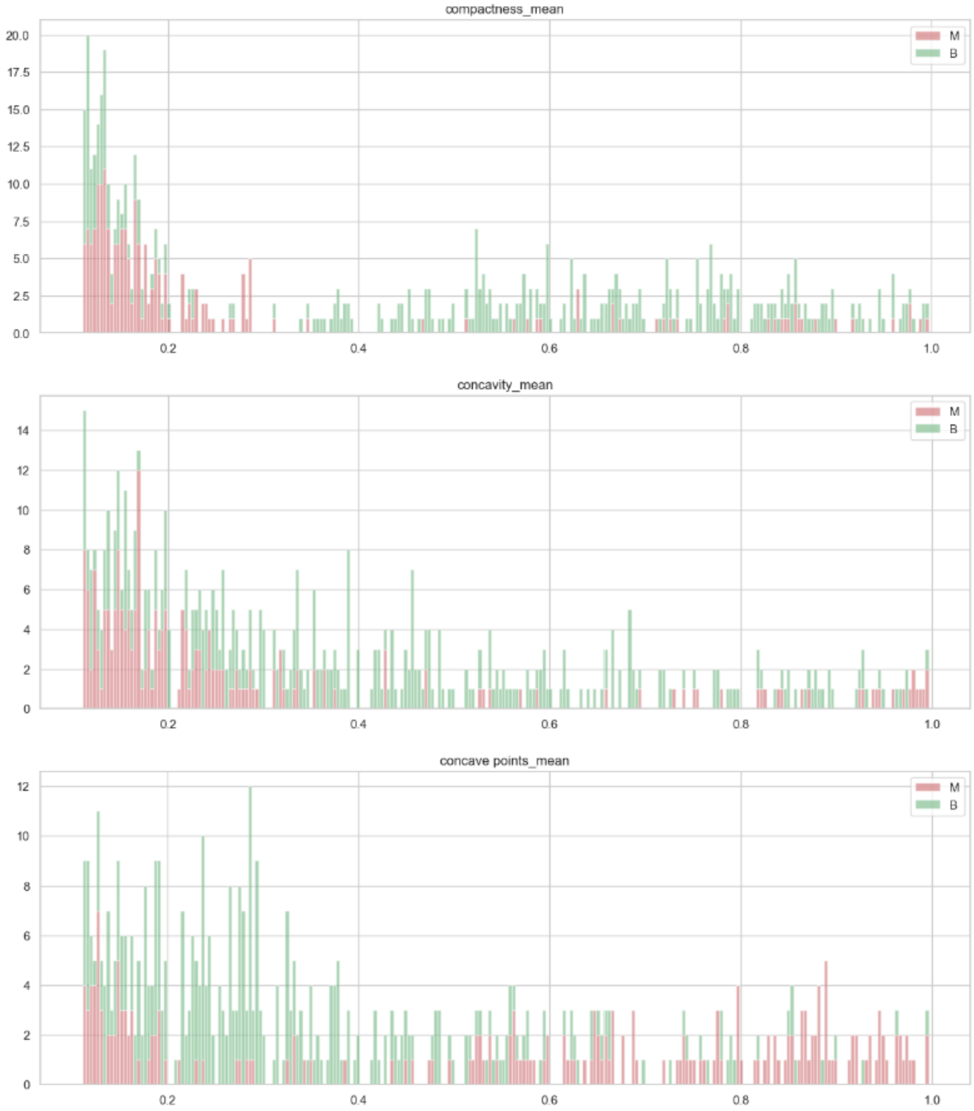


Figure 21:Histogram 3

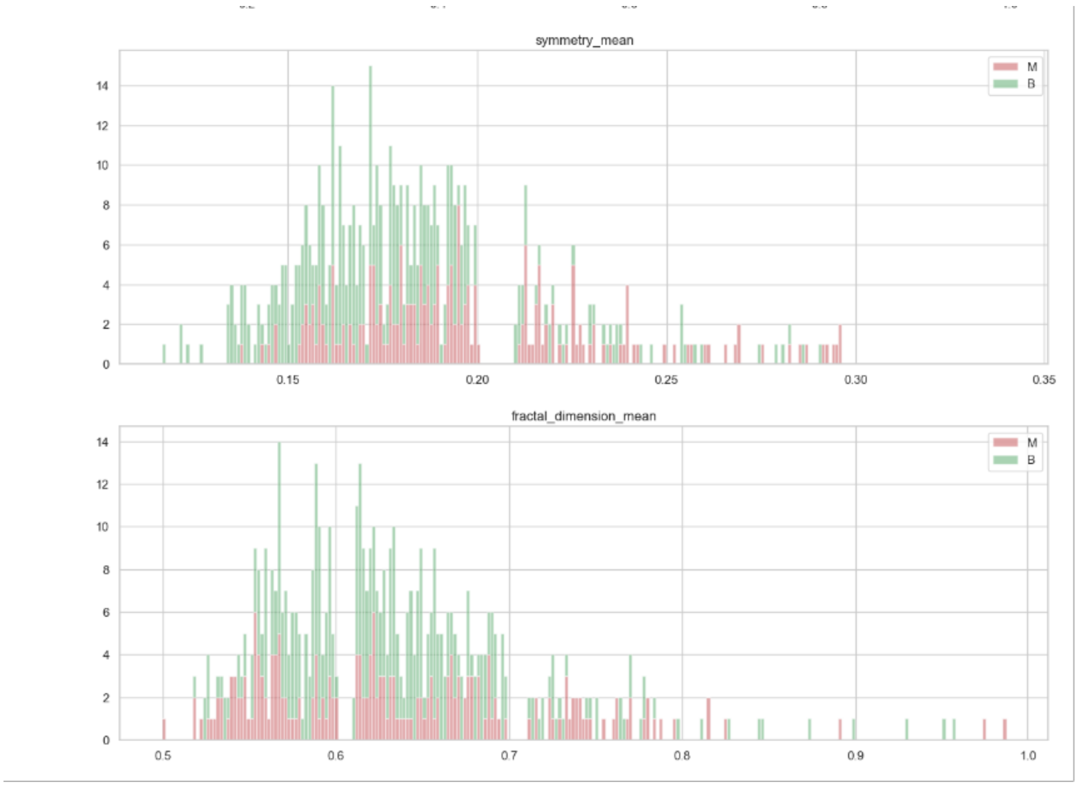
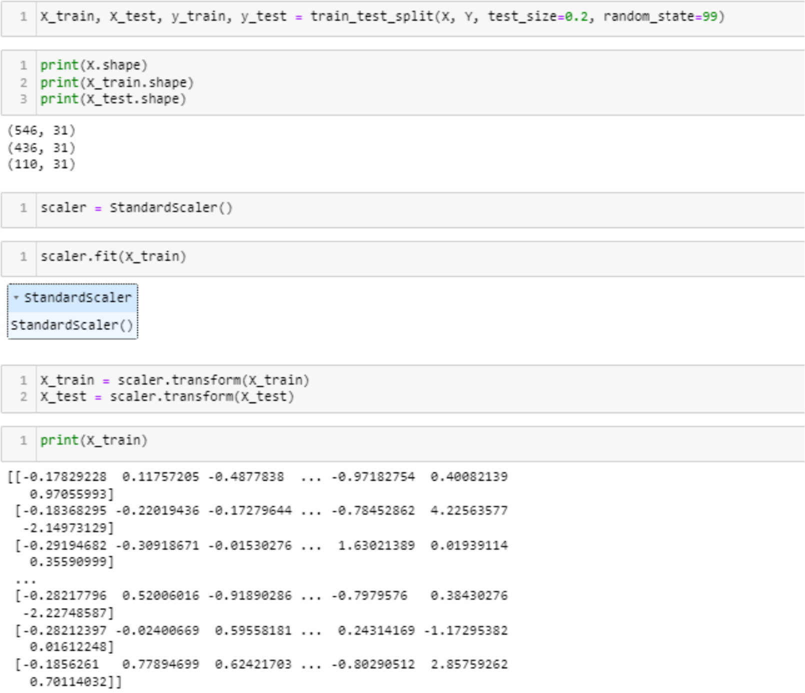


Figure 22:Histogram 4

The provided code aims to visually compare the distribution of selected features between malignant (M) and benign (B) diagnoses in a breast cancer dataset. Firstly, the data is filtered to create a subset ('dataBenign') containing only benign cases. Subsequently, histograms are generated for each specified feature, depicting the distribution of values for both malignant and benign cases. The subplots, organized in a 10-row by 1-column grid, allow for a side-by-side comparison of the feature distributions. The histograms are stacked, with different colours (red for malignant and green for benign) aiding in differentiation. The bin width is dynamically calculated for each feature to optimize visualization. Legends are included for clarity, indicating which color corresponds to each diagnosis. This visualization provides insights into the potential differences in feature distributions between benign and malignant cases, contributing to the initial exploratory data analysis in the model development process.

3.7. Data Pre-processing

Data pre-processing is an essential stage in the data mining process, encompassing the transformation of raw data into a more interpretable format. This step is instrumental in preparing the data for analysis and machine learning applications. Various data pre-processing techniques are employed to clean and refine the data, ensuring that it is in a suitable format for utilization by machine learning algorithms. By addressing issues such as missing values, handling categorical variables, and scaling numerical features, data pre-processing enhances the quality and compatibility of the data for subsequent modelling, ultimately contributing to the effectiveness of machine learning algorithms.



```
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=99)
```

```
print(X.shape)
print(X_train.shape)
print(X_test.shape)
```

```
(546, 31)
(436, 31)
(110, 31)
```

```
scaler = StandardScaler()
```

```
scaler.fit(X_train)
```

```
▾ StandardScaler
StandardScaler()
```

```
X_train = scaler.transform(X_train)
X_test = scaler.transform(X_test)
```

```
print(X_train)
```

```
[[-0.17829228  0.11757205 -0.4877838  ... -0.97182754  0.40082139
   0.97055993]
 [-0.18368295 -0.22019436 -0.17279644 ... -0.78452862  4.22563577
  -2.14973129]
 [-0.29194682 -0.30918671 -0.01530276 ...  1.63021389  0.01939114
   0.35590999]
 ...
 [-0.28217796  0.52006016 -0.91890286 ... -0.7979576   0.38430276
  -2.22748587]
 [-0.28212397 -0.02400669  0.59558181 ...  0.24314169 -1.17295382
   0.01612248]
 [-0.1856261   0.77894699  0.62421703 ... -0.80290512  2.85759262
   0.70114032]]
```

Figure 23:Data Pre-processing

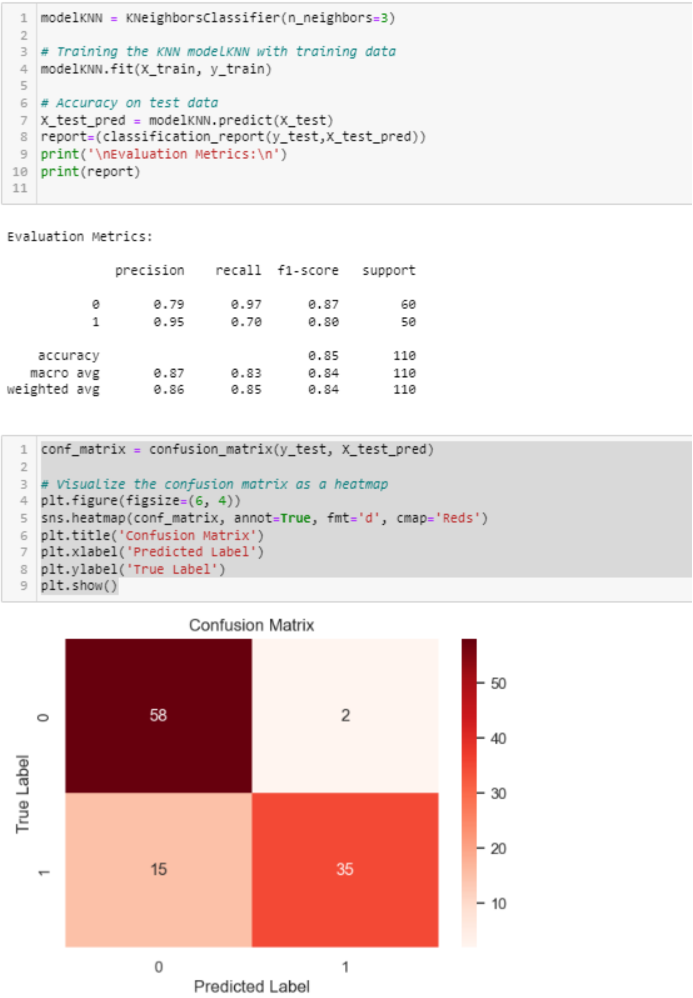
The provided code is part of the typical process of preparing data for machine learning. Initially, it uses the train_test_split function from scikit-learn to split the dataset represented by X (features) and Y (target variable) into training and testing sets. The parameter test_size=0.2 indicates that 20% of the data will be reserved for testing. The random_state=99 ensures reproducibility in the data split. Subsequently, the code prints the shapes of the original feature matrix (X) and the newly created training (X_train) and testing (X_test) sets. Following this, a StandardScaler from scikit-learn is employed to standardize the feature values based on the training set. The scaler is fit to the training data using scaler.fit(X_train), and then both the training and testing sets are transformed to have standardized features using X_train = scaler.transform(X_train) and X_test = scaler.transform(X_test). Standardization is crucial for ensuring that all features have a similar scale, which aids in the convergence and performance of many machine learning algorithms. The standardized training set is then printed for inspection.

3.8. Machine Learning model evaluation and prediction

The dataset will now be trained and tested with three different machine learning algorithms using the training and testing datasets created above.

3.8.1. KNN Classifier

Training and testing the model using KNN Classifier:



```
modelKNN = KNeighborsClassifier(n_neighbors=3)

# Training the KNN modelKNN with training data
modelKNN.fit(X_train, y_train)

# Accuracy on test data
X_test_pred = modelKNN.predict(X_test)
report=(classification_report(y_test,X_test_pred))
print('\nEvaluation Metrics:\n')
print(report)
```

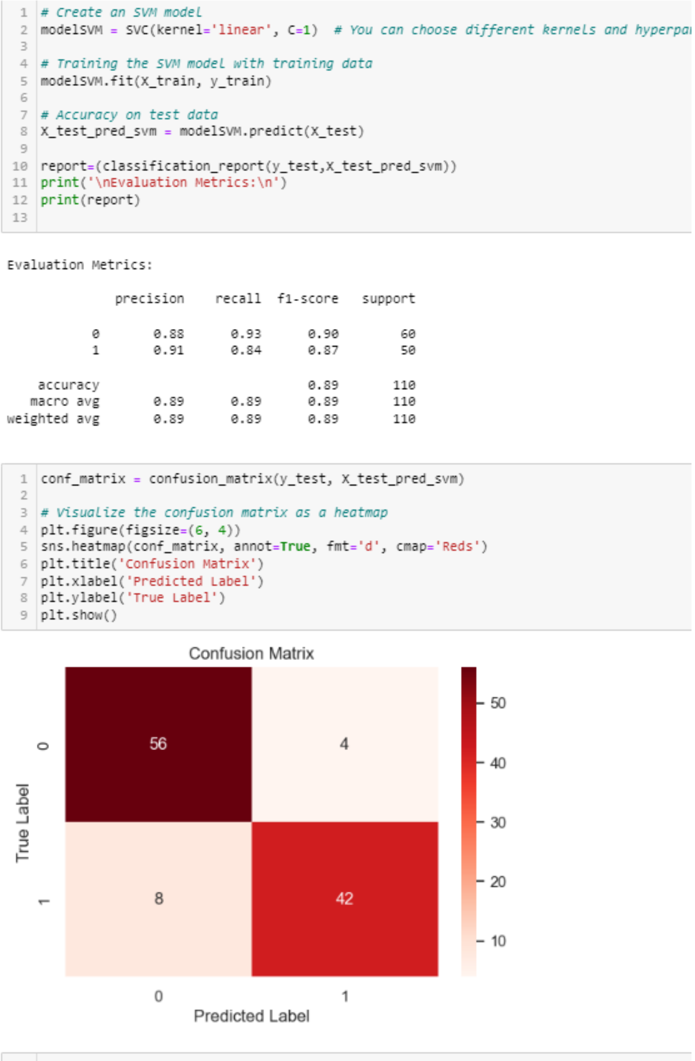
```
Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.79      0.97      0.87        60
           1       0.95      0.70      0.80        50

    accuracy                           0.85       110
   macro avg       0.87      0.83      0.84       110
weighted avg       0.86      0.85      0.84       110
```

```
conf_matrix = confusion_matrix(y_test, X_test_pred)

# Visualize the confusion matrix as a heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Reds')
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

Figure 24:Knn Classifier accuracy score and confusion matrix

3.8.2. Supply Vector Machine (SVM)

Training and testing the model using Supply Vector Machine (SVM):



```
# Create an SVM model
modelSVM = SVC(kernel='linear', C=1)  # You can choose different kernels and hyperpa

# Training the SVM model with training data
modelSVM.fit(X_train, y_train)

# Accuracy on test data
X_test_pred_svm = modelSVM.predict(X_test)

report=(classification_report(y_test,X_test_pred_svm))
print('\nEvaluation Metrics:\n')
print(report)

```

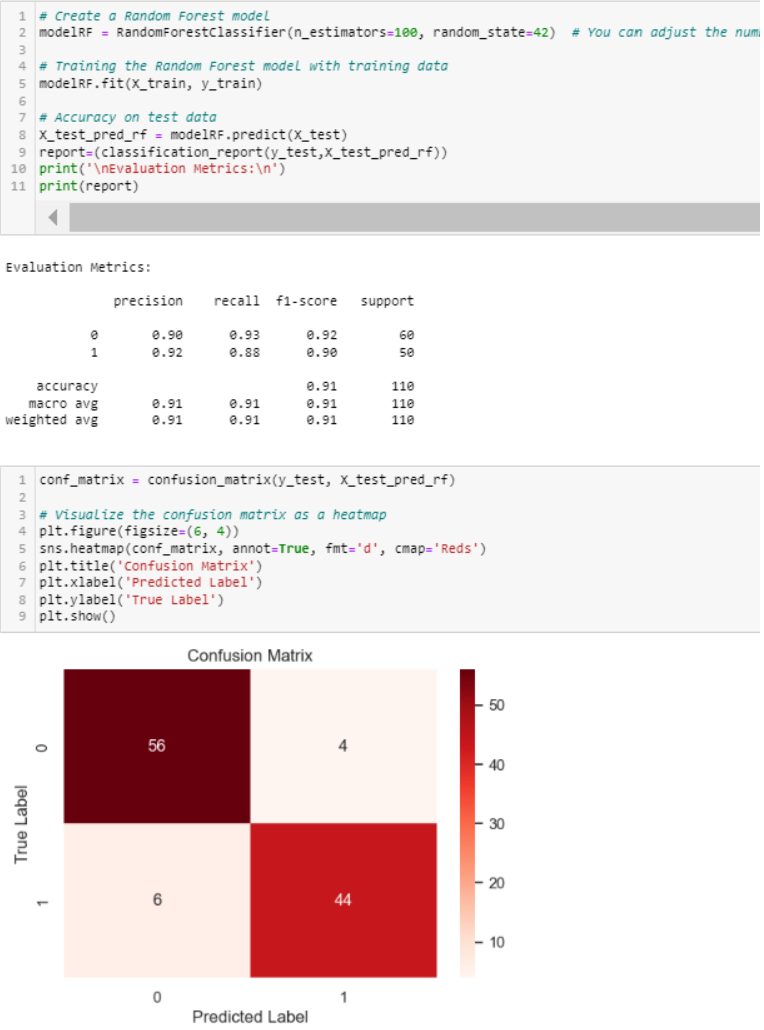
```
Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.88      0.93      0.90        60
           1       0.91      0.84      0.87        50

    accuracy                           0.89       110
   macro avg       0.89      0.89      0.89       110
weighted avg       0.89      0.89      0.89       110
```

```
conf_matrix = confusion_matrix(y_test, X_test_pred_svm)

# Visualize the confusion matrix as a heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Reds')
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

Figure 25:SVM accuracy score and confusion matrix

3.8.3. Random Forest Classifier

Training and testing the model using Random Forest Classifier:



```
# Create a Random Forest model
modelRF = RandomForestClassifier(n_estimators=100, random_state=42)  # You can adjust the num

# Training the Random Forest model with training data
modelRF.fit(X_train, y_train)

# Accuracy on test data
X_test_pred_rf = modelRF.predict(X_test)
report=(classification_report(y_test,X_test_pred_rf))
print('\nEvaluation Metrics:\n')
print(report)
```

```
Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.90      0.93      0.92        60
           1       0.92      0.88      0.90        50

    accuracy                           0.91       110
   macro avg       0.91      0.91      0.91       110
weighted avg       0.91      0.91      0.91       110
```

```
conf_matrix = confusion_matrix(y_test, X_test_pred_rf)

# Visualize the confusion matrix as a heatmap
plt.figure(figsize=(6, 4))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Reds')
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

Figure 26:Random Forest accuracy score and confusion matrix

3.8.4. Comparison between algorithms

Three algorithms have been utilized to train and test the model. The accuracy scores of these algorithms will be compared, and the algorithm demonstrating the highest accuracy will be selected for further processing, specifically hyperparameter tuning.

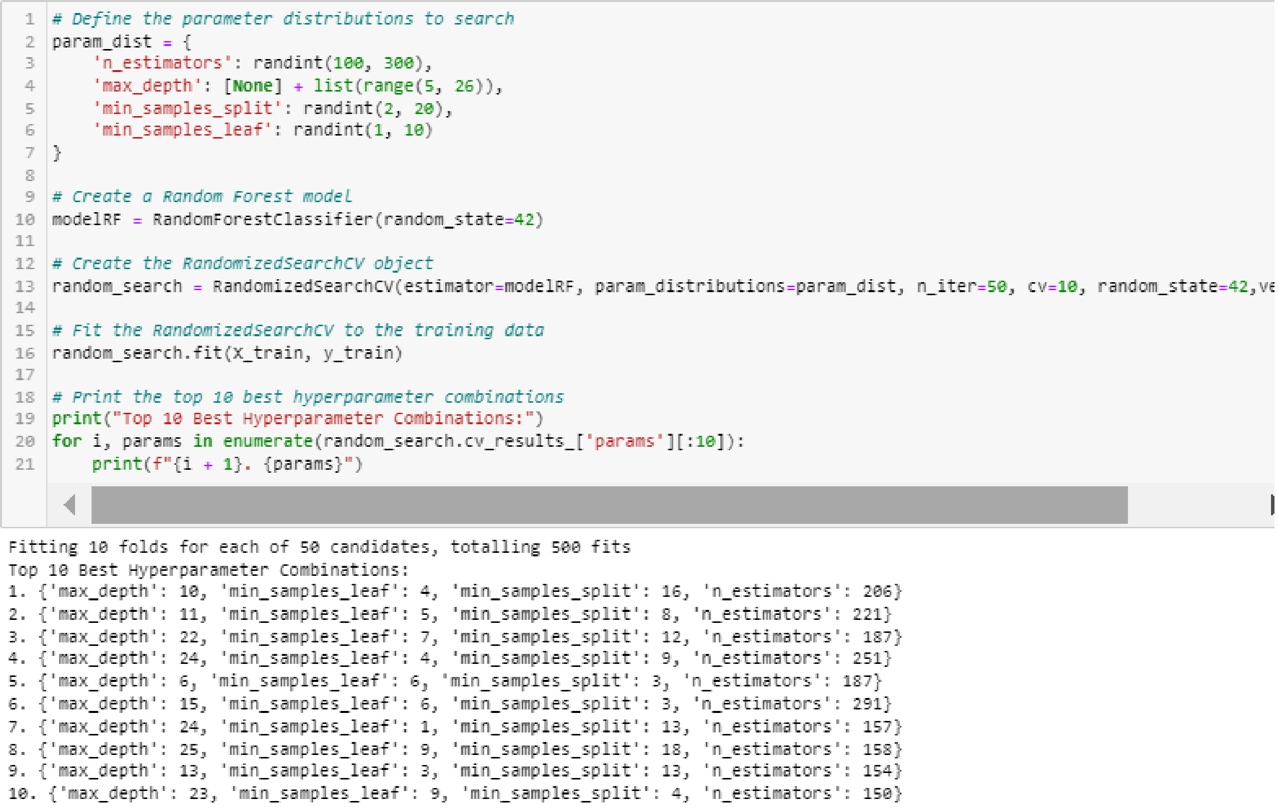
Algorithm	Accuracy
KNN Classifier	85%
Supply Vector Machine (SVM)	89%
Random Forest Classifier	91%

Table 1: Comparison of accuracy among the tables

From the above comparison table, it is observed that the Random Forest classifier has the highest accuracy score among the three algorithms chosen for training and testing the dataset.

3.9. Best Parameters

Identification of optimal hyperparameters for tuning will be accomplished through the utilization of RandomizedSearchCV, as illustrated below.

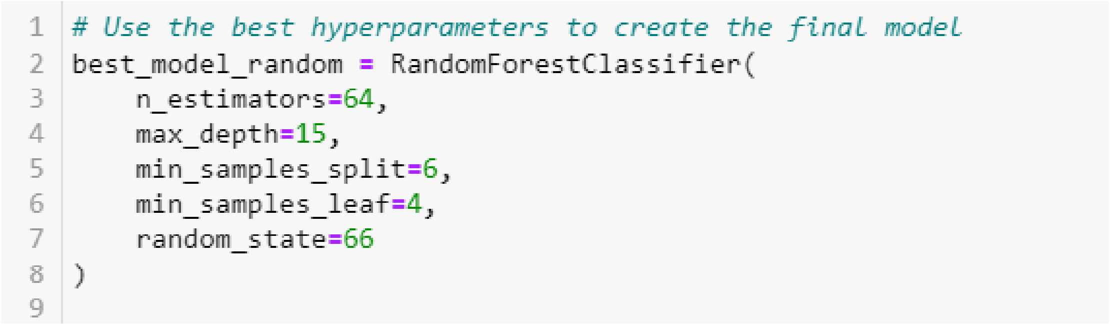


```
# Define the parameter distributions to search
param_dist = {
    'n_estimators': randint(100, 300),
    'max_depth': [None] + list(range(5, 26)),
    'min_samples_split': randint(2, 20),
    'min_samples_leaf': randint(1, 10)
}

# Create a Random Forest model
modelRF = RandomForestClassifier(random_state=42)

# Create the RandomizedSearchCV object
random_search = RandomizedSearchCV(estimator=modelRF, param_distributions=param_dist, n_iter=50, cv=10, random_state=42,ve

# Fit the RandomizedSearchCV to the training data
random_search.fit(X_train, y_train)

# Print the top 10 best hyperparameter combinations
print("Top 10 Best Hyperparameter Combinations:")
for i, params in enumerate(random_search.cv_results_['params'][:10]):
    print(f"{i + 1}. {params}")
```

```
Fitting 10 folds for each of 50 candidates, totalling 500 fits
Top 10 Best Hyperparameter Combinations:
1. {'max_depth': 10, 'min_samples_leaf': 4, 'min_samples_split': 16, 'n_estimators': 206}
2. {'max_depth': 11, 'min_samples_leaf': 5, 'min_samples_split': 8, 'n_estimators': 221}
3. {'max_depth': 22, 'min_samples_leaf': 7, 'min_samples_split': 12, 'n_estimators': 187}
4. {'max_depth': 24, 'min_samples_leaf': 4, 'min_samples_split': 9, 'n_estimators': 251}
5. {'max_depth': 6, 'min_samples_leaf': 6, 'min_samples_split': 3, 'n_estimators': 187}
6. {'max_depth': 15, 'min_samples_leaf': 6, 'min_samples_split': 3, 'n_estimators': 291}
7. {'max_depth': 24, 'min_samples_leaf': 1, 'min_samples_split': 13, 'n_estimators': 157}
8. {'max_depth': 25, 'min_samples_leaf': 9, 'min_samples_split': 18, 'n_estimators': 158}
9. {'max_depth': 13, 'min_samples_leaf': 3, 'min_samples_split': 13, 'n_estimators': 154}
10. {'max_depth': 23, 'min_samples_leaf': 9, 'min_samples_split': 4, 'n_estimators': 150}
```

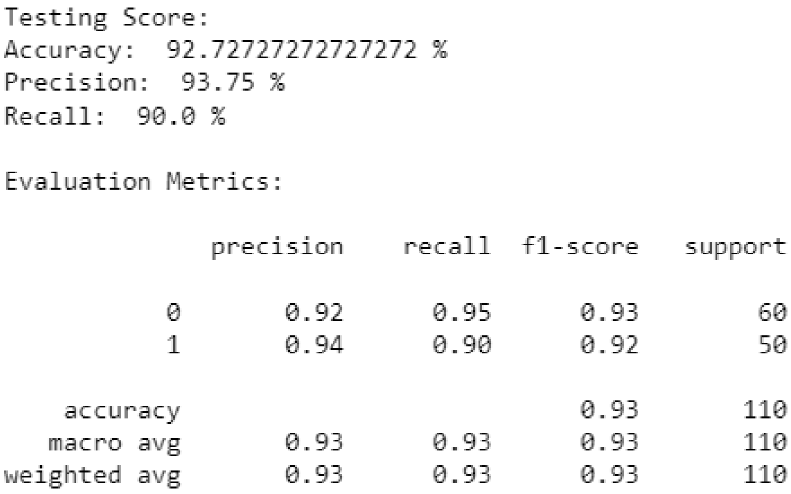
Figure 27:Identifying Best Parameters

3.10. Hyperparameter Tuning the Random Forest Classifier Model



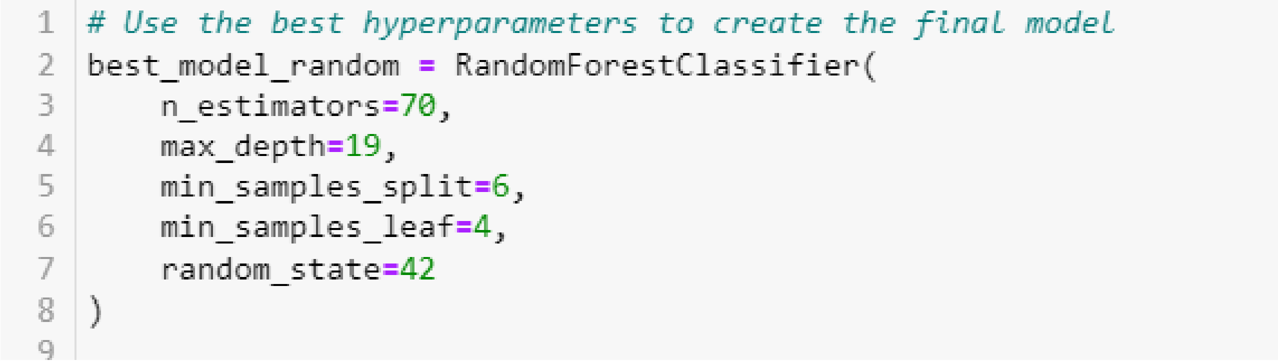
```python
# Use the best hyperparameters to create the final model
best_model_random = RandomForestClassifier(
    n_estimators=64,
    max_depth=15,
    min_samples_split=6,
    min_samples_leaf=4,
    random_state=66
)

```

Figure 28:Input 1



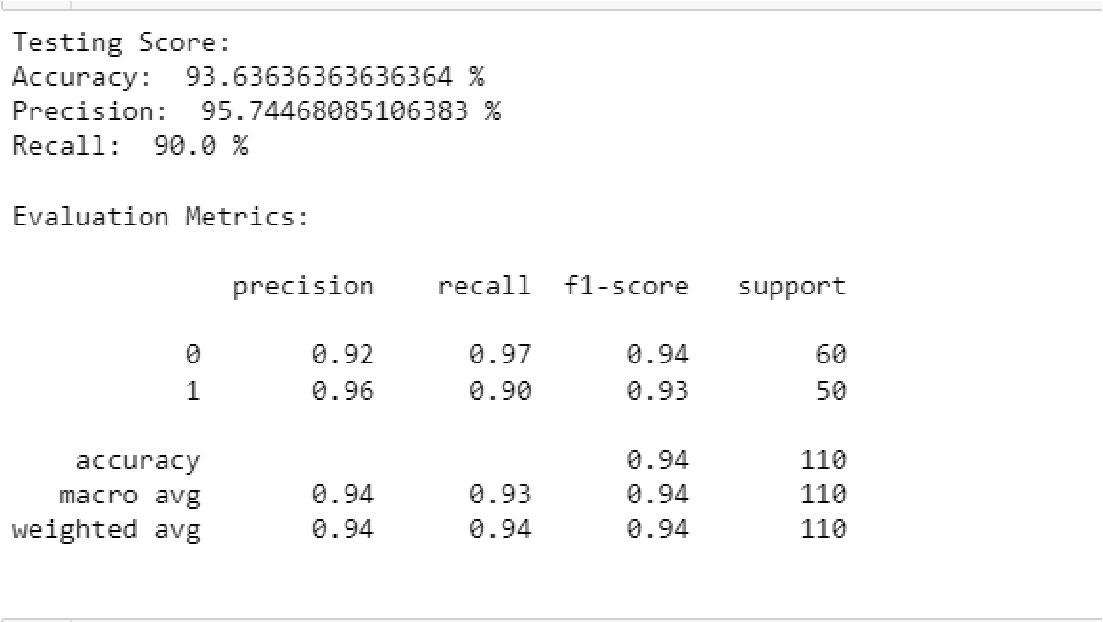
```
Testing Score:
Accuracy:  92.72727272727272 %
Precision:  93.75 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.95      0.93        60
           1       0.94      0.90      0.92        50

    accuracy                           0.93       110
   macro avg       0.93      0.93      0.93       110
weighted avg       0.93      0.93      0.93       110
```

Figure 29:Output 1



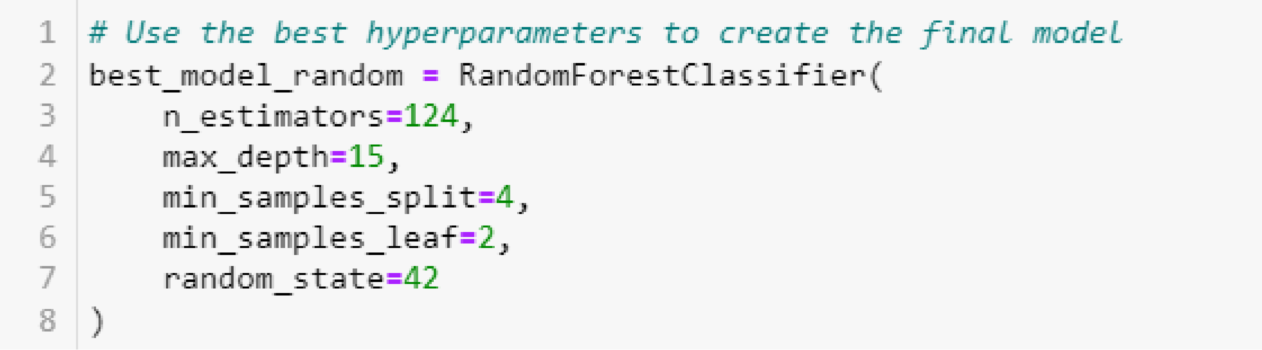
```python
# Use the best hyperparameters to create the final model
best_model_random = RandomForestClassifier(
    n_estimators=70,
    max_depth=19,
    min_samples_split=6,
    min_samples_leaf=4,
    random_state=42
)
```

Figure 30:Input 2



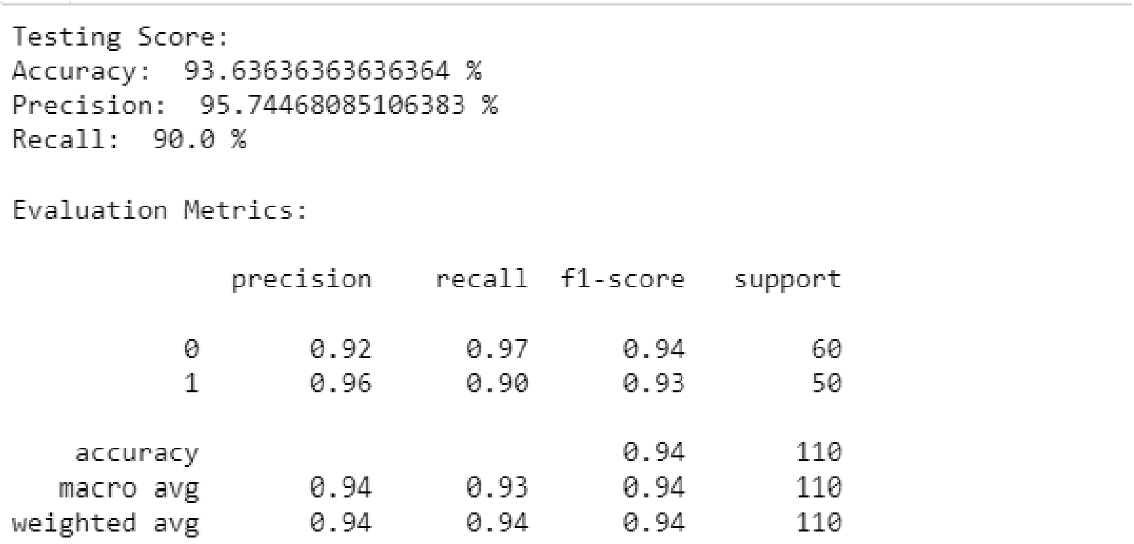
```
Testing Score:
Accuracy:  93.63636363636364 %
Precision:  95.74468085106383 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.97      0.94        60
           1       0.96      0.90      0.93        50

    accuracy                           0.94       110
   macro avg       0.94      0.93      0.94       110
weighted avg       0.94      0.94      0.94       110
```

Figure 31:Output 2



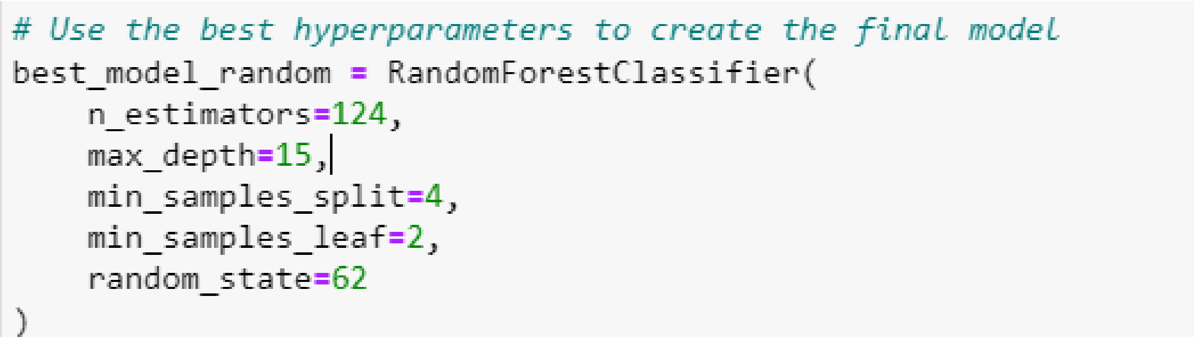
```
# Use the best hyperparameters to create the final model
best_model_random = RandomForestClassifier(
    n_estimators=124,
    max_depth=15,
    min_samples_split=4,
    min_samples_leaf=2,
    random_state=42
)
```

Figure 32:Input 3



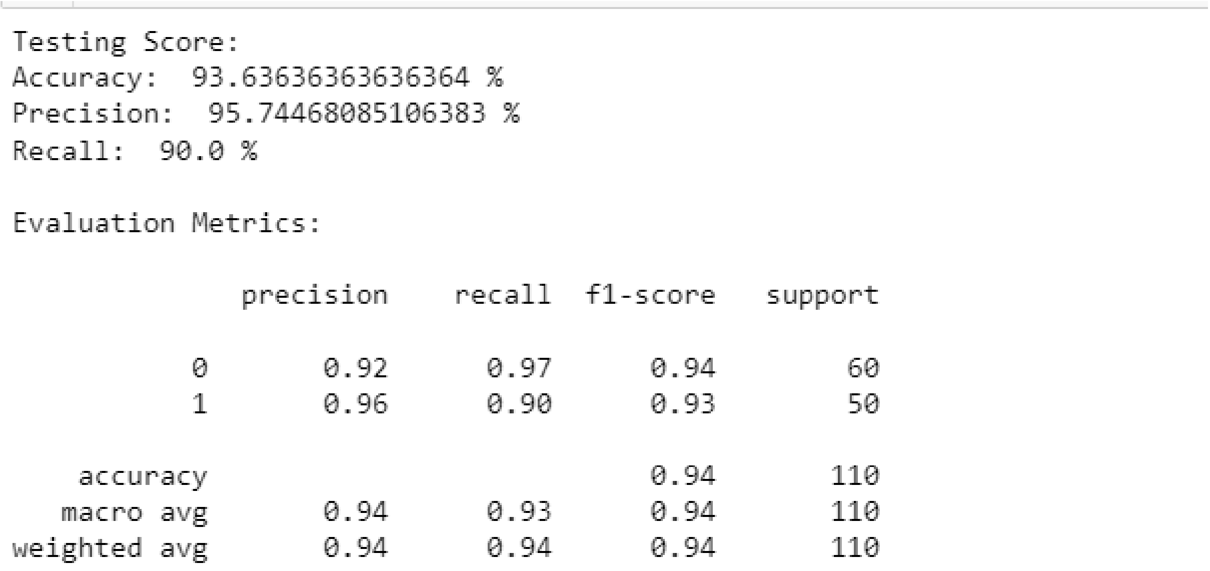
```
Testing Score:
Accuracy:  93.63636363636364 %
Precision:  95.74468085106383 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.97      0.94        60
           1       0.96      0.90      0.93        50

    accuracy                           0.94       110
   macro avg       0.94      0.93      0.94       110
weighted avg       0.94      0.94      0.94       110
```

Figure 33:Output 3



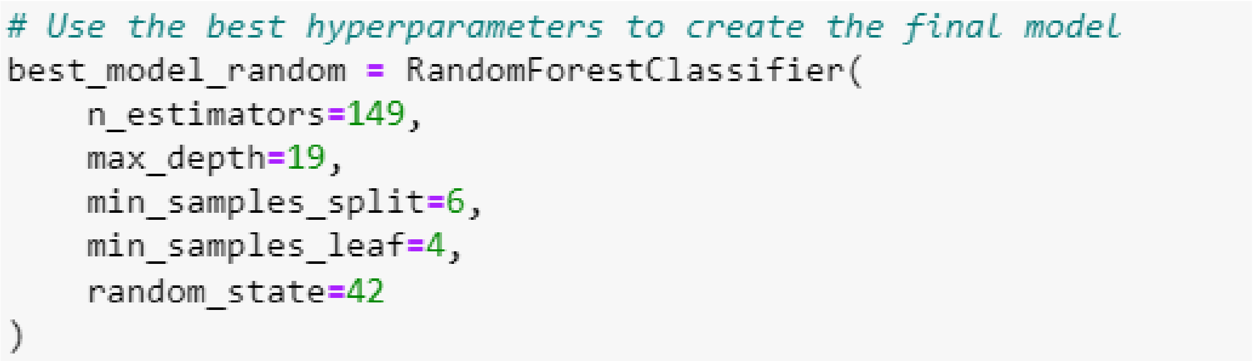
```
# Use the best hyperparameters to create the final model
best_model_random = RandomForestClassifier(
    n_estimators=124,
    max_depth=15,
    min_samples_split=4,
    min_samples_leaf=2,
    random_state=62
)
```

Figure 34:Input 4



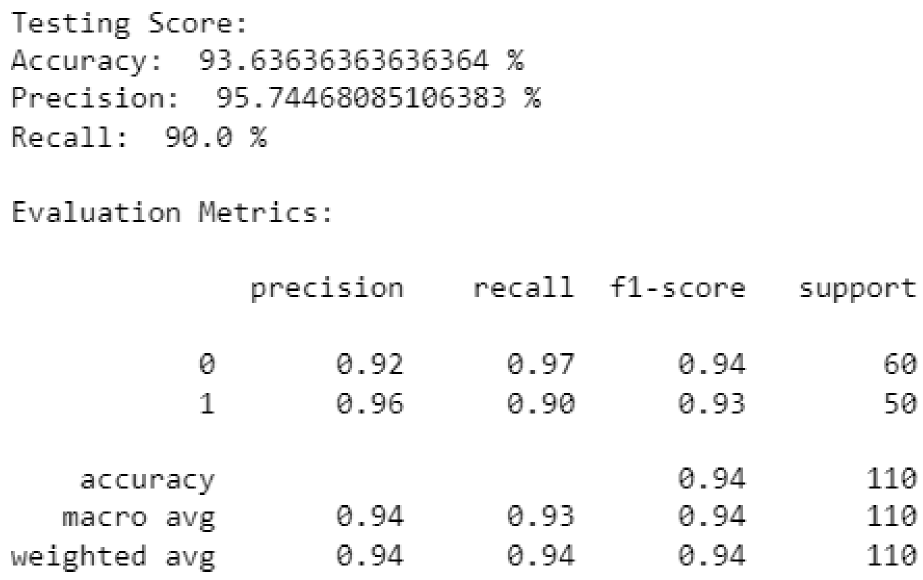
```
Testing Score:
Accuracy:  93.63636363636364 %
Precision:  95.74468085106383 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.97      0.94        60
           1       0.96      0.90      0.93        50

    accuracy                           0.94       110
   macro avg       0.94      0.93      0.94       110
weighted avg       0.94      0.94      0.94       110
```

Figure 35:Output 4



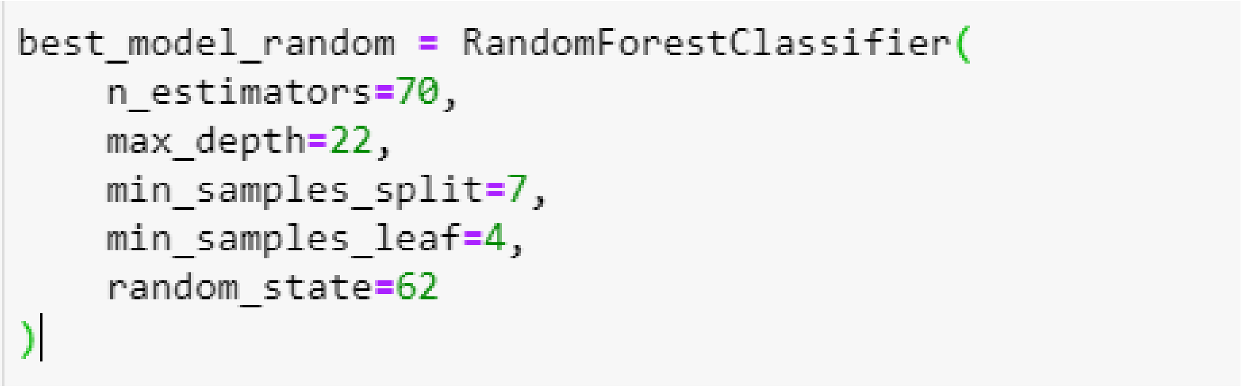
```
# Use the best hyperparameters to create the final model
best_model_random = RandomForestClassifier(
    n_estimators=149,
    max_depth=19,
    min_samples_split=6,
    min_samples_leaf=4,
    random_state=42
)
```

Figure 36:Input 5



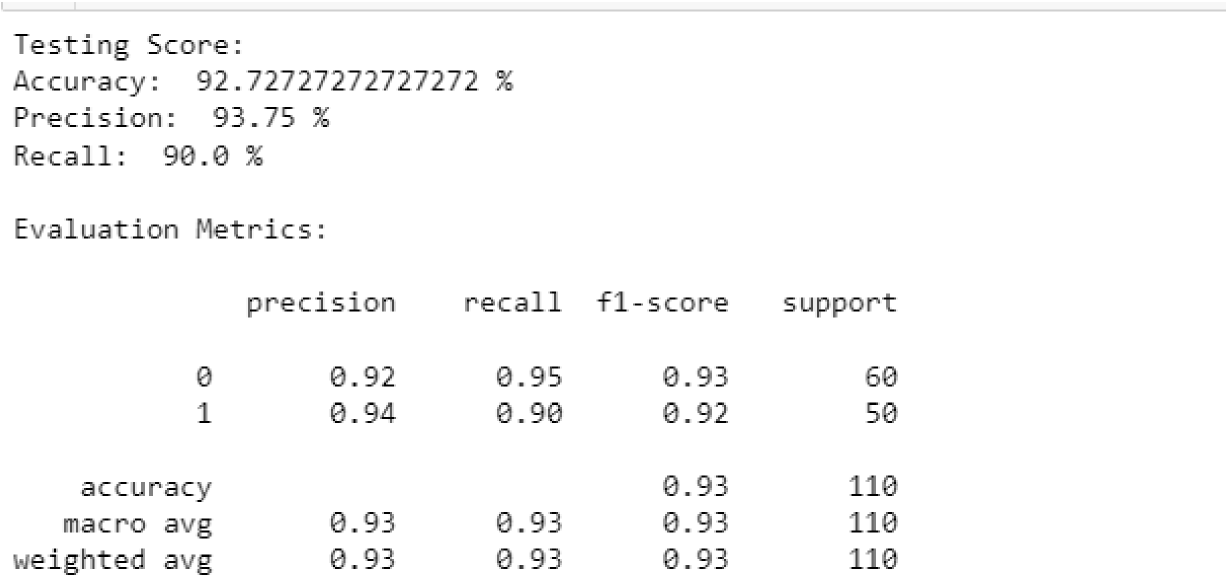
```
Testing Score:
Accuracy:  93.63636363636364 %
Precision:  95.74468085106383 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.97      0.94        60
           1       0.96      0.90      0.93        50

    accuracy                           0.94       110
   macro avg       0.94      0.93      0.94       110
weighted avg       0.94      0.94      0.94       110
```

Figure 37:Output 5



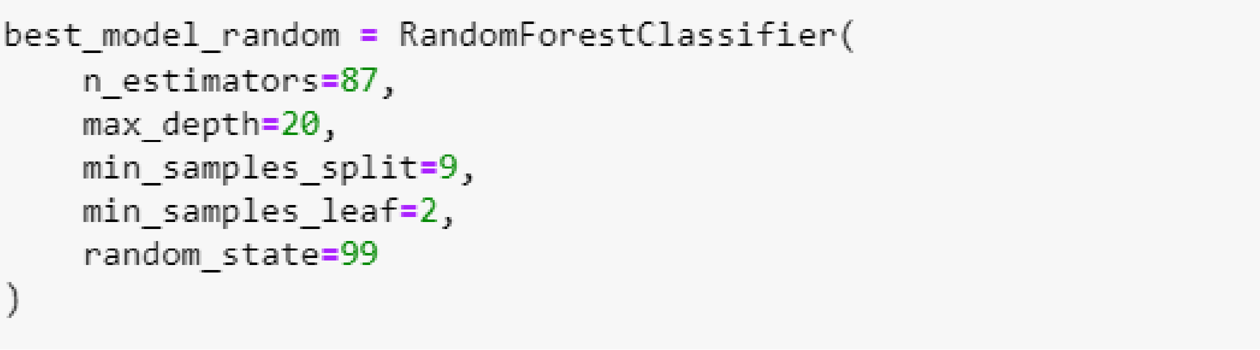
```
best_model_random = RandomForestClassifier(
    n_estimators=70,
    max_depth=22,
    min_samples_split=7,
    min_samples_leaf=4,
    random_state=62
)
```

Figure 38:Input 6



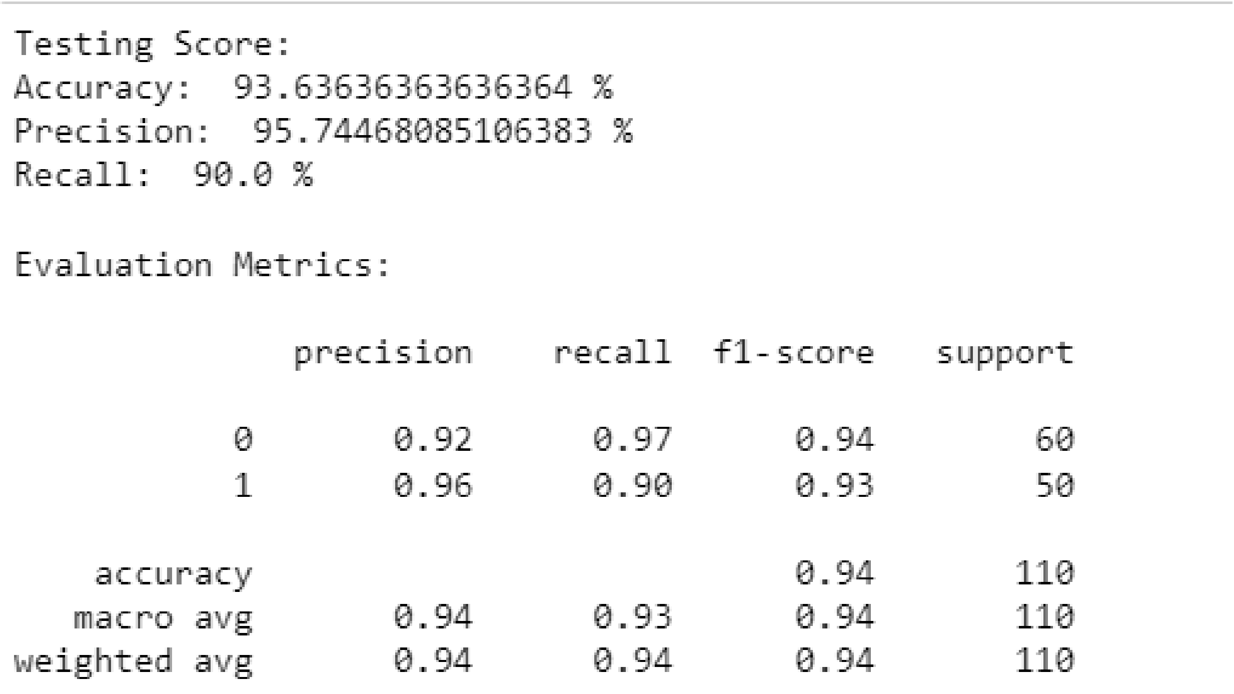
```
Testing Score:
Accuracy:  92.72727272727272 %
Precision:  93.75 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.95      0.93        60
           1       0.94      0.90      0.92        50

    accuracy                           0.93       110
   macro avg       0.93      0.93      0.93       110
weighted avg       0.93      0.93      0.93       110
```

Figure 39:Output 6



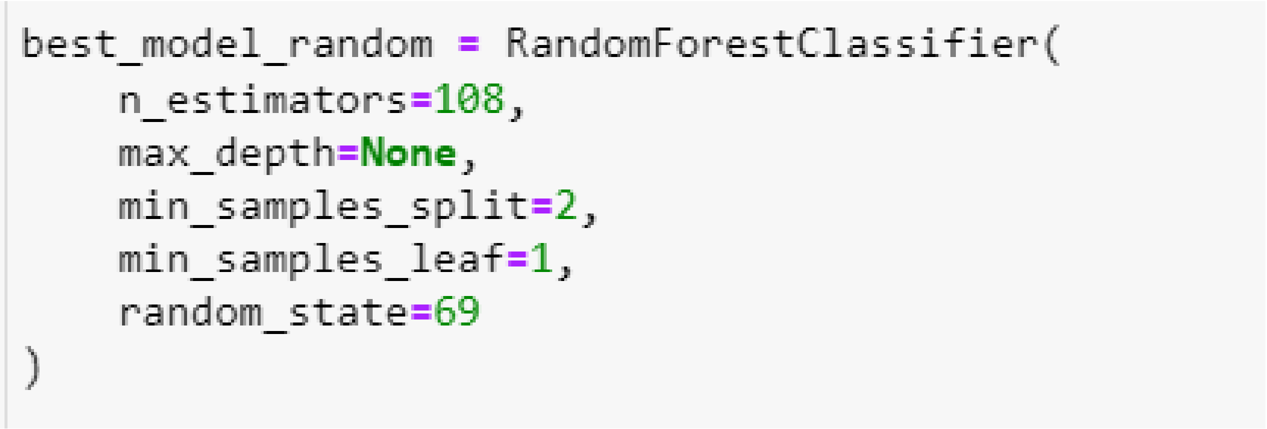
```
best_model_random = RandomForestClassifier(
    n_estimators=87,
    max_depth=20,
    min_samples_split=9,
    min_samples_leaf=2,
    random_state=99
)
```

Figure 40:Input 7



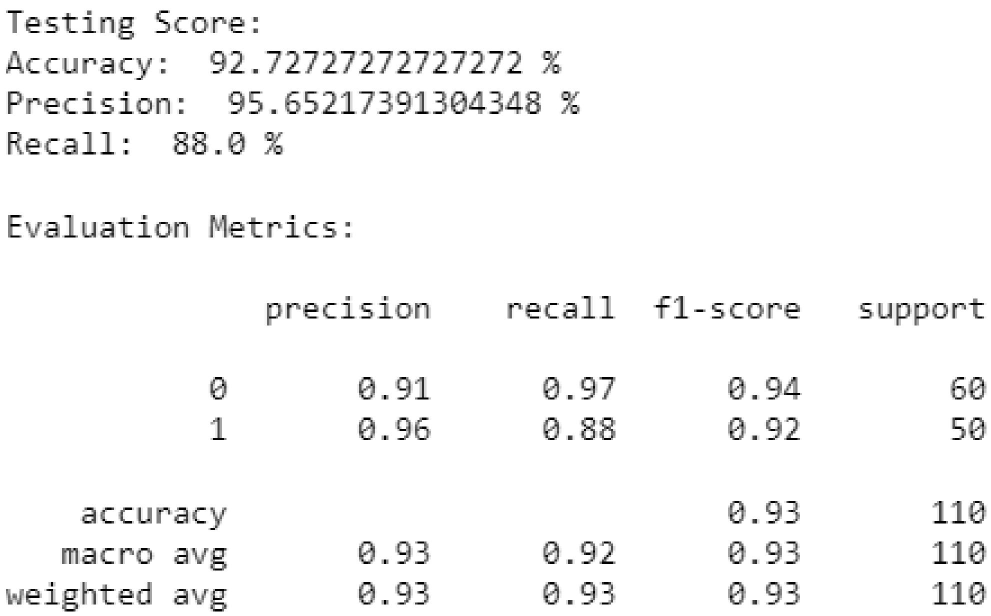
```
Testing Score:
Accuracy:  93.63636363636364 %
Precision:  95.74468085106383 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.97      0.94        60
           1       0.96      0.90      0.93        50

    accuracy                           0.94       110
   macro avg       0.94      0.93      0.94       110
weighted avg       0.94      0.94      0.94       110
```

Figure 41:Output 7



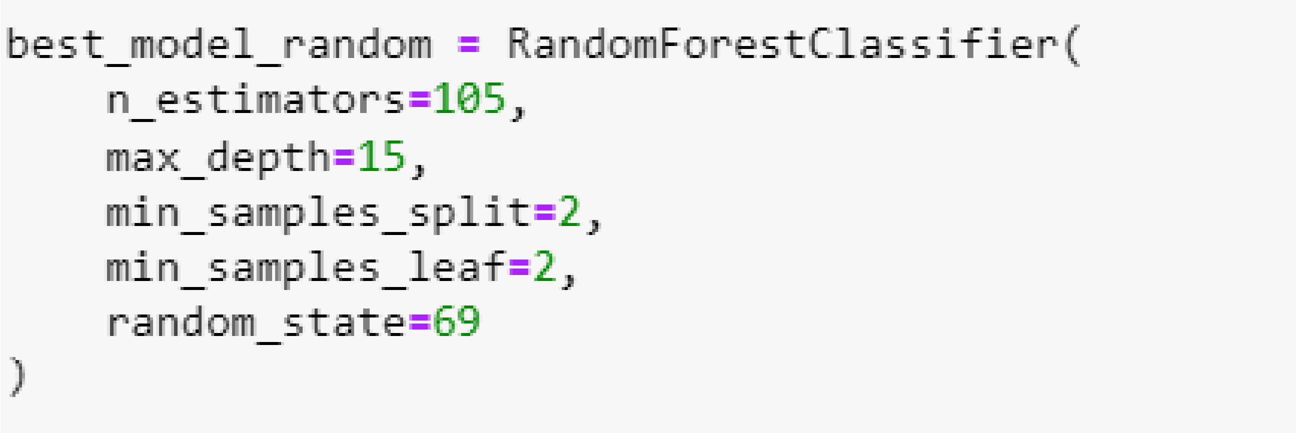
```
best_model_random = RandomForestClassifier(
    n_estimators=108,
    max_depth=None,
    min_samples_split=2,
    min_samples_leaf=1,
    random_state=69
)
```

Figure 42:Input 8



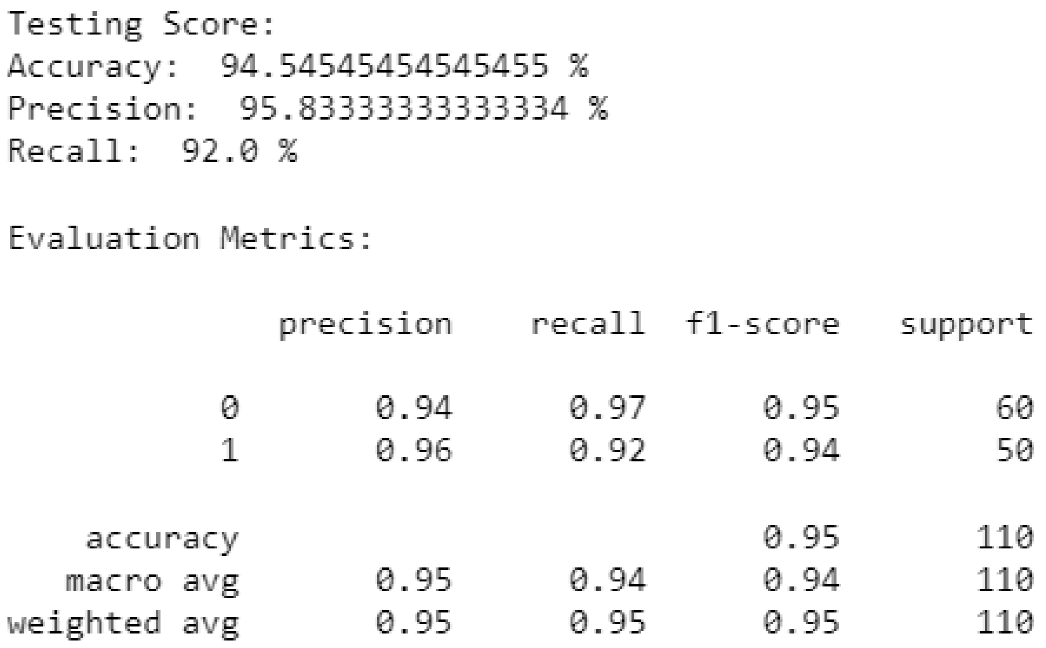
```
Testing Score:
Accuracy:  92.72727272727272 %
Precision:  95.65217391304348 %
Recall:  88.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.91      0.97      0.94        60
           1       0.96      0.88      0.92        50

    accuracy                           0.93       110
   macro avg       0.93      0.92      0.93       110
weighted avg       0.93      0.93      0.93       110
```

Figure 43:Output 8



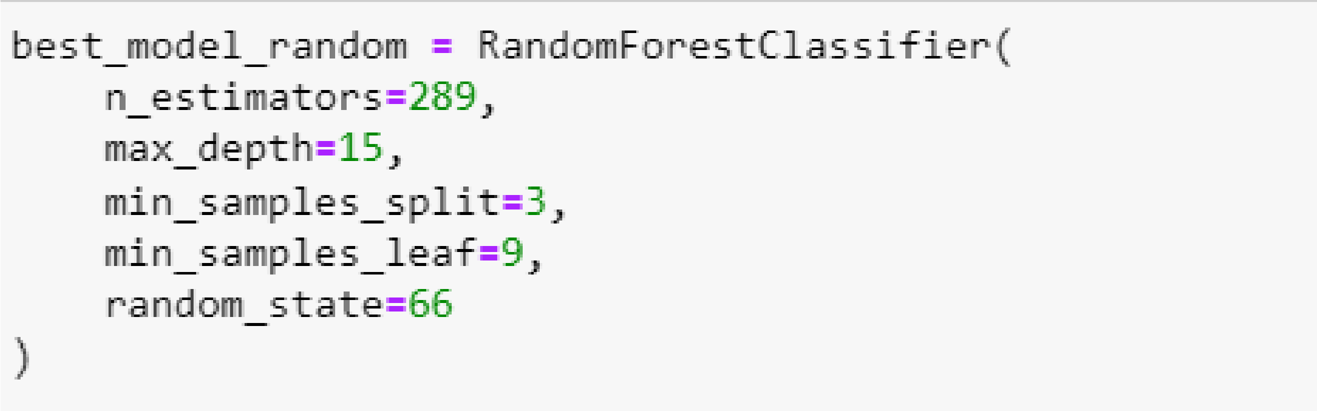
```
best_model_random = RandomForestClassifier(
    n_estimators=105,
    max_depth=15,
    min_samples_split=2,
    min_samples_leaf=2,
    random_state=69
)
```

Figure 44:Input 9



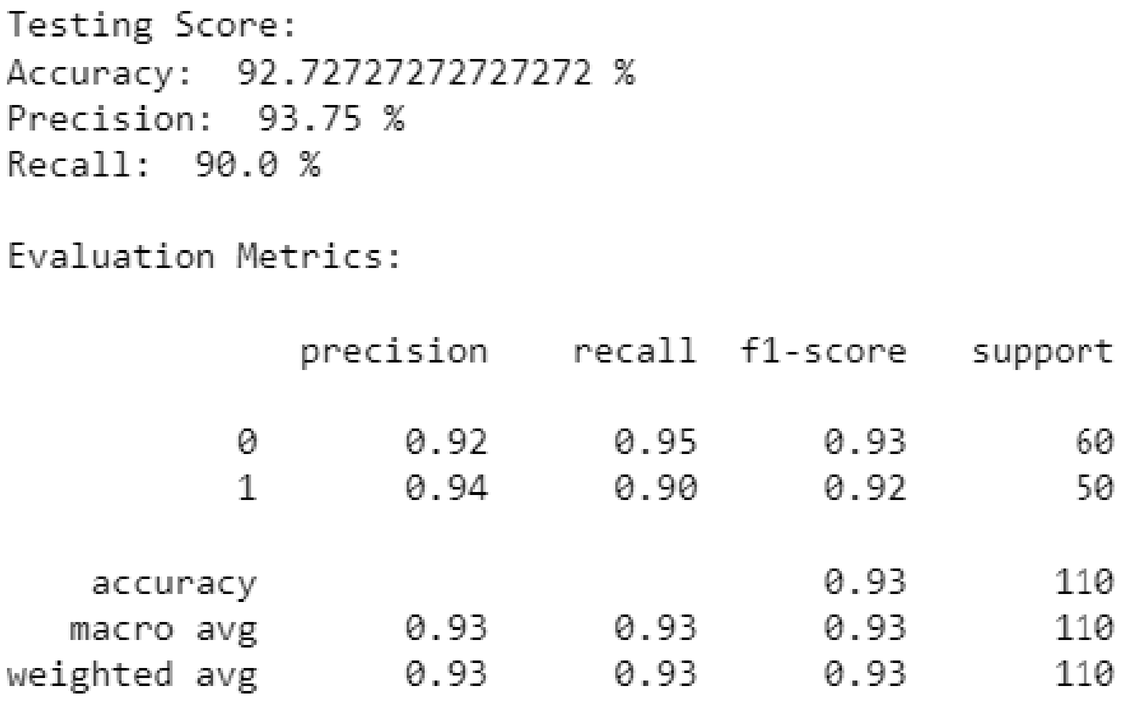
```
Testing Score:
Accuracy:  94.54545454545455 %
Precision:  95.83333333333334 %
Recall:  92.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.94      0.97      0.95        60
           1       0.96      0.92      0.94        50

    accuracy                           0.95       110
   macro avg       0.95      0.94      0.94       110
weighted avg       0.95      0.95      0.95       110
```

Figure 45:Output 9



```
best_model_random = RandomForestClassifier(
    n_estimators=289,
    max_depth=15,
    min_samples_split=3,
    min_samples_leaf=9,
    random_state=66
)
```

Figure 46:Input 10



```
Testing Score:
Accuracy:  92.72727272727272 %
Precision:  93.75 %
Recall:  90.0 %

Evaluation Metrics:

              precision    recall  f1-score   support

           0       0.92      0.95      0.93        60
           1       0.94      0.90      0.92        50

    accuracy                           0.93       110
   macro avg       0.93      0.93      0.93       110
weighted avg       0.93      0.93      0.93       110
```

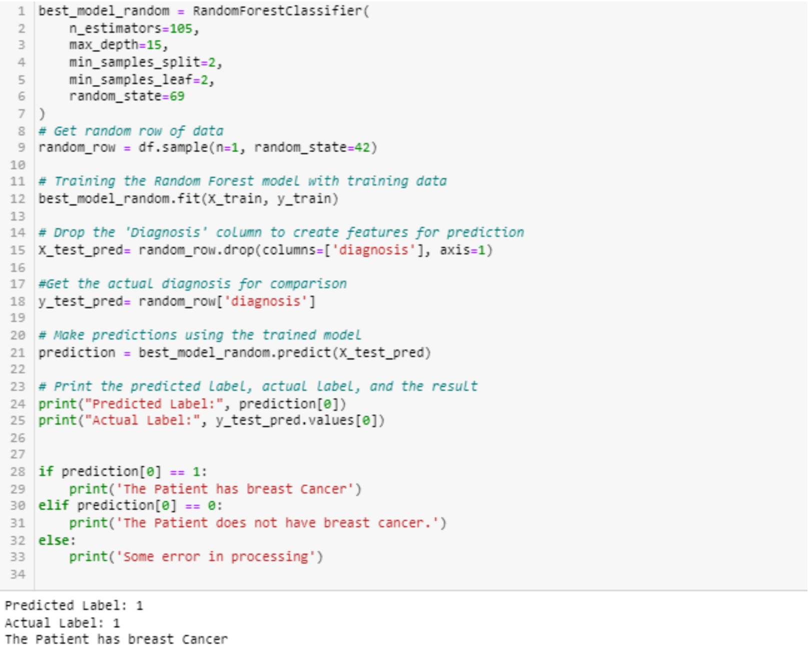
Figure 47:Output 10

After the hyperparameter tuning, the best the combination of n_estimator=105, max_depth=15, Min_sample_split=2, min_sample_leaf=2 gives the best precision, recall, and accuracy among the tested combinations.

N_estimators	Max_depth	Min_sample_split	Min_sample_leaf	Random_state	Precision	Recall	Accuracy
64	15	6	4	66	0.94	0.90	0.93
70	19	6	4	42	0.96	0.90	0.94
124	15	4	2	42	0.96	0.90	0.94
128	10	4	2	42	0.96	0.90	0.94
149	19	6	4	42	0.96	0.90	0.94
70	22	7	4	62	0.94	0.90	0.93
87	20	9	2	99	0.96	0.90	0.94
108	None	2	1	69	0.96	0.88	0.93
105	15	2	2	69	0.96	0.92	0.95
289	15	3	9	66	0.94	0.90	0.93

Table 2:Comparison after hyperparameter tuning

After the hyperparameter tuning, the model is tested with random data to find out if the patient has cancer or not. The result is shown below.



```python
best_model_random = RandomForestClassifier(
    n_estimators=105,
    max_depth=15,
    min_samples_split=2,
    min_samples_leaf=2,
    random_state=69
)
# Get random row of data
random_row = df.sample(n=1, random_state=42)

# Training the Random Forest model with training data
best_model_random.fit(X_train, y_train)

# Drop the 'Diagnosis' column to create features for prediction
X_test_pred= random_row.drop(columns=['diagnosis'], axis=1)

#Get the actual diagnosis for comparison
y_test_pred= random_row['diagnosis']

# Make predictions using the trained model
prediction = best_model_random.predict(X_test_pred)

# Print the predicted label, actual label, and the result
print("Predicted Label:", prediction[0])
print("Actual Label:", y_test_pred.values[0])


if prediction[0] == 1:
    print('The Patient has breast Cancer')
elif prediction[0] == 0:
    print('The Patient does not have breast cancer.')
else:
    print('Some error in processing')

```

```
Predicted Label: 1
Actual Label: 1
The Patient has breast Cancer
```

Figure 48: Prediction Testing

4. Conclusion

4.1. Analysis of the work done

In conclusion, this project has served as a valuable opportunity to delve into the practical applications and expansive potentials of Artificial Intelligence and Machine Learning in real world scenarios. Extensive research was conducted across various problem domains feasible for AI and ML applications. Among the explored domains, the focus was directed towards leveraging AI and ML techniques for breast cancer prediction, aligning closely with the module's learning objectives.

Thorough investigation was carried out within the breast cancer prediction domain, encompassing a review of existing literature and methodologies employed by professionals. This comprehensive review laid the groundwork, offering insights into prevalent approaches used to address this critical issue. Following the assessment, the Random Forest algorithm was identified as a suitable candidate for implementation within this project, owing to its consistent performance across similar studies.

The project's solution strategy was meticulously planned and detailed through initial documentation stages, including pseudocode and flowcharts. These documents facilitated a high-level understanding of the entire process, delineating each step of the solution comprehensively. This structured approach aided in strategizing the execution of the proposed breast cancer prediction system.

4.2. How the solution addresses real world problems

The development and application of breast cancer prediction systems can significantly address real-world problems in the management and treatment of breast cancer. These systems provide a means to predict the likelihood of a woman developing breast cancer based on various factors, such as demographic, laboratory, and mammographic risk factors.

The potential for early detection of breast cancer through these systems can lead to more effective treatment options and potentially better outcome for patients. This early detection can also reduce the number of unnecessary screenings and tests, thereby reducing financial and psychological burden for patients. Machine learning models have shown high accuracy in predicting breast cancer, which can further improve the effectiveness of these systems. Breast cancer prediction systems can also help optimize the allocation of healthcare resources. By predicting the risk of breast cancer, healthcare providers can prioritize screenings and interventions for high-risk individuals, ensuring that resources are used most effectively. Furthermore, these systems can provide personalized care by identifying risk factors unique to each individual. This can help healthcare providers tailor treatment plans to the individual’s unique situation, potentially improving treatment outcomes. Lastly, some breast cancer prediction systems can also predict the risk of reassurance, which is a critical factor in long-term care and treatment planning.

In summary, breast cancer prediction systems can address real-world problems by enabling early detection, reducing overdiagnosis, improving accuracy, optimizing resource allocation, providing personalized care, and predicting recurrence risk. These systems have the potential to significantly improve the management and treatment of breast cancer.

4.3. Limitations of the system

The Wisconsin dataset and a Random Forest classifier may encounter several limitations when used to build a breast cancer prediction system. The size of the dataset, which includes a relatively small number of observations, could be insufficient for training a robust machine learning model, leading to overfitting. Furthermore, if the dataset exhibits an imbalanced distribution of malignant and benign cases, the model may become biased towards the majority class, resulting in high accuracy rates but poor predictive performance on the minority class, which is often the class of greater clinical importance. The effectiveness of the Random Forest algorithm heavily relies on the relevance of the input features, and features that do not contribute to the predictive power of the model can introduce noise and lead to less accurate predictions. Highly correlated features can lead to redundancy in the model. Model complexity can also pose a challenge, as Random Forest models can become quite complex with many trees and deep decision paths, making the model difficult to interpret. Computational resources can also be a constraint, especially when the prediction system needs to be deployed in a resource-constrained environment. Lastly, the model may not generalize well to other populations due to differences in demographics, genetics, and other factors. No model is perfect, and misclassifications can have serious implications in medical diagnosis. Therefore, thorough exploratory data analysis, feature engineering, hyperparameter tuning, and validation using techniques such as cross-validation are recommended to mitigate these limitations.

Bibliography

altexoft, 2021. *alexsoft.* [Online] Available at: https://www.altexsoft.com/blog/unsupervised-machine-learning/ [Accessed 15 January 2024].

Arslan, K. et al., 2023. *Breast Cancer Detection and Prevention Using Machine Learning,* s.l.: s.n.

aslanyan, T., 2023. *freecodecamp.* [Online] Available at: https://www.freecodecamp.org/news/machine-learning-handbook/ [Accessed 16 January 2024].

B.J.Copeland, 2024. *Britannica.* [Online] Available at: https://www.britannica.com/technology/artificial-intelligence [Accessed 14 January 2024].

Bo, Z., Huiping, S. & Hongtao, W., 2023. *Machine Learning and AI in Cancer Prognosis, Prediction, and Treatment Selection: A Critical Approach,* s.l.: PMC.

Brownlee, J., 2023. *machinelearningmaster.* [Online] Available at: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/ [Accessed 16 January 2024].

Cleveland Clinic, 2024. *clevelandclinic.org.* [Online] Available at: https://my.clevelandclinic.org/health/diseases/3986-breast-cancer [Accessed 16 January 2024].

DatabaseTown, 2024. *DatabaseTown.* [Online] Available at: https://databasetown.com/unsupervised-learning-types-applications/ [Accessed 15 January 2024].

David, D., 2022. *Random Forest Classifier Tutorial: How to Use Tree-Based Algorithms for Machine Learning.* [Online] Available at: https://www.freecodecamp.org/news/how-to-use-the-tree-based-algorithm-for-machine-learning/

Donges, N., 2023. *builtin.* [Online] Available at: https://builtin.com/data-science/random-forest-algorithm [Accessed 16 January 2024].

Driscoll, M., 2023. *Jupyter Notebook: An Introduction.* [Online] Available at: https://realpython.com/jupyter-notebook-introduction/ [Accessed 17 January 2024].

enjoy algorithms, 2024. *enjoy algorithms.* [Online] Available at: https://www.enjoyalgorithms.com/blogs/supervised-unsupervised-and-semisupervised-learning

IBM, 2024. *ibm.* [Online] Available at: https://www.ibm.com/topics/supervised-learning [Accessed 15 January 2024].

IBM, 2024. *K-Nearest Neighbors Algorithm.* [Online] Available at: https://www.ibm.com/topics/knn [Accessed 16 January 2024].

IBM, 2024. *Random Forest.* [Online] Available at: https://www.ibm.com/topics/random-forest [Accessed 16 january 2024].

Islan, R., Awan, R., Bakar, S. A. & Monira, I., 2019. *A Precise Breast Cancer Detection Approach Using Ensemble of Random Forest with AdaBoost,* s.l.: IEEE.

javatpoint, 2024. *javatpoint.* [Online] Available at: https://www.javatpoint.com/supervised-machine-learning

laskowski, N., 2024. *techtarget.* [Online] Available at: https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence
[Accessed 14 January 2024].

National Breast Cancer Foundation, 2024. *Breast Tumors.* [Online] Available at: https://www.nationalbreastcancer.org/breast-tumors/
[Accessed 16 January 2024].

Oliviera, G., 2019. *ResearchGate.* [Online] Available at: https://www.researchgate.net/publication/335638493_Encoder-Decoder_Methods_for_Semantic_Segmentation_Efficiency_and_Robustness_Aspects

Reza, R. et al., 2022. *Prediction of Breast Cancer using Machine Learning Approaches,* s.l.: J Biomed Phys End.

Saini, A., 2024. *Analytucs Vidhya.* [Online] Available at: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/
[Accessed 16 January 2024].

Santa Clara University , 2024. *9 Real-Life Examples of Reinforcement Learning.* [Online] Available at: https://onlinedegrees.scu.edu/media/blog/9-examples-of-reinforcement-learning
[Accessed 15 January 2024].

Sarker, I. H., 2021. *Machine Learning: Algorithms, Real-World Applications and Research Directions,* s.l.: Springer Link.

Schroer, A., 2024. *builtin.* [Online] Available at: https://builtin.com/artificial-intelligence
[Accessed 14 January 2024].

seidor , 2021. *BASIC CONCEPTS OF ARTIFICIAL INTELLIGENCE.* [Online] Available at: https://opentrends.net/en/article/basic-concepts-artificial-intelligence
[Accessed 14 January 2024].

Seraydarian, L., 2023. *What Is Prediction in ML and Why Is It Important?.* [Online] Available at: https://plat.ai/blog/machine-learning-prediction/

Sharmin, A. & Das Annesha, D. A., 2021. *Malignant and Benign Breast Cancer Classification using Machine Learning Algorithms,* s.l.: IEEE.

Sodhi, P., Awasthi, N. & Sharma, V., 2019. Introduction to Machine Learning and Its Basic Application in. *Introduction to Machine Learning and Its Basic Application in,* p. 22.

Srivastava, T., 2024. *analyticsvidhya.* [Online]
Available at: https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/
[Accessed 15 January 2024].

Sypnosis, 2024. *Sypnosis.* [Online]
Available at: https://www.synopsys.com/ai/what-is-reinforcement-learning.html
[Accessed 15 January 2024].