UNIVERSITY SCHOOL OF AUTOMATION AND ROBOTICS
GURU GOBIND SINGH INDRAPRASTHA UNIVERSITY
EAST DELHI CAMPUS, SURAJMAL VIHAR, DELHI- 110032

Summer Training Report

On

Classification On Breast Cancer Wisconsin (Diagnostic)

Submitted in partial fulfillment of the

requirements for the completion of one month's summer internship/training [ART 355]

Name: **B Dhruv**

Enrollment Number **03519011921**

Under the supervision of

Dr. Anirban Dandapat

DECLARATION

I hereby declare that the Summer Training Report entitled **Classification On Breast Cancer Wisconsin (Diagnostic)** is an authentic record of work completed as requirements of Summer Training (ART 355) during the period from 01/08/2023 to 30/08/23 in University School of Automation and Robotics/CDAC/NIC/DRDO/PEC/etc under the supervision of **Dr. Anirban Dandapat.**

(Signature of student)

B Dhruv

Enrollment No: 03519011921

Date: _____

(Signature of Supervisor)

Dr. Anirban Dandapat

Date: _____

Acknowledgement

I would like to express my sincere gratitude to Dr. Anirban Dandapat for his invaluable guidance and unwavering support throughout my internship at GGSIPU East Campus (USAR). His expertise, patience, and encouragement have been instrumental in shaping my understanding and enhancing my skills. I am truly thankful for his mentorship, which has been a constant source of inspiration during this learning journey. I am also grateful to GGSIPU East Campus (USAR) for providing me with the opportunity to work under his supervision, allowing me to gain practical experience and insights that have been invaluable for my professional growth.

I would like to express my sincere gratitude to the UCI Machine Learning Repository for providing the Breast Cancer Wisconsin (Diagnostic) dataset used in this machine learning project. The availability of this data was paramount in conducting the analysis and developing the models presented in this report. The UCI Machine Learning Repository's dedication to collecting and sharing valuable datasets has been instrumental in advancing research and learning in the field of machine learning. Proper attribution to the UCI Machine Learning Repository, accessible at [https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic], is essential in acknowledging their efforts, as this dataset significantly contributed to the outcomes of this project.

Tables of Content

List of figures

Abbreviations

1. FNB: Fine Needle Biopsy
2. UCI: University of California, Irvine
3. ID: Identification Number
4. SVC: Support Vector Classification
5. NB: Naive Bayes
6. GPU: Graphics Processing Unit
7. CPU: Central Processing Unit
8. RAM: Random Access Memory
9. IDE: Integrated Development Environment
10. NP: NumPy
11. PD: Pandas
12. PLT: Matplotlib
13. SB: Seaborn
14. SKL: Scikit-learn

Abstract

This report presents a comprehensive analysis of breast cancer diagnosis utilizing machine learning techniques on the Breast Cancer Wisconsin (Diagnostic) dataset. Breast cancer is a prevalent reproductive malignancy affecting women globally, necessitating accurate and timely diagnosis for improved prognosis and treatment outcomes. Fine Needle Biopsy (FNB) has emerged as a less invasive alternative to traditional diagnostic procedures, prompting the need for automated diagnostic systems.

The study employs diverse classification algorithms, including Decision Trees, Random Forest, Support Vector Classification (SVC), Logistic Regression, and Gaussian Naive Bayes (NB), to predict breast cancer presence based on cellular characteristics derived from 569 image samples. The dataset, consisting of 32 attributes, encapsulates morphological features, such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension, reported in three forms: mean, standard error, and "worst." The analysis reveals that the Random Forest classifier achieves an impressive accuracy rate of 97.90%, outperforming other algorithms.

A predictive system utilizing the Random Forest classifier has been constructed to facilitate real-world applications in breast cancer diagnosis. This system offers a valuable tool for medical practitioners and researchers, enhancing early detection and patient care. The study underscores the significance of machine learning in improving breast cancer diagnosis, ultimately contributing to better patient outcomes in the fight against this formidable disease.

Introduction

Breast cancer, a formidable and prevalent reproductive malignancy primarily impacting women, is characterized by the abnormal growth of tissues within the breast, often presenting as nipple discharge, lumps, or changes in skin texture around the nipple area. The diagnosis of breast cancer has traditionally involved invasive surgical procedures such as full biopsies, which entail significant discomfort and risks. However, a less invasive technique known as Fine Needle Biopsy (FNB) has emerged, allowing for the examination of small tissue samples from tumors.

Breast cancer holds global significance, affecting women predominantly within the age range of 25 to 50 years. Its complex etiology is attributed to a combination of genetic, hormonal, lifestyle, and environmental factors. Familial gene mutations, obesity, aging, hormonal abnormalities after menopause, and other variables contribute to the multifaceted nature of this disease. Despite these factors, breast cancer lacks a definitive prevention mechanism, underscoring the critical importance of early detection for improving prognosis and reducing treatment expenses.

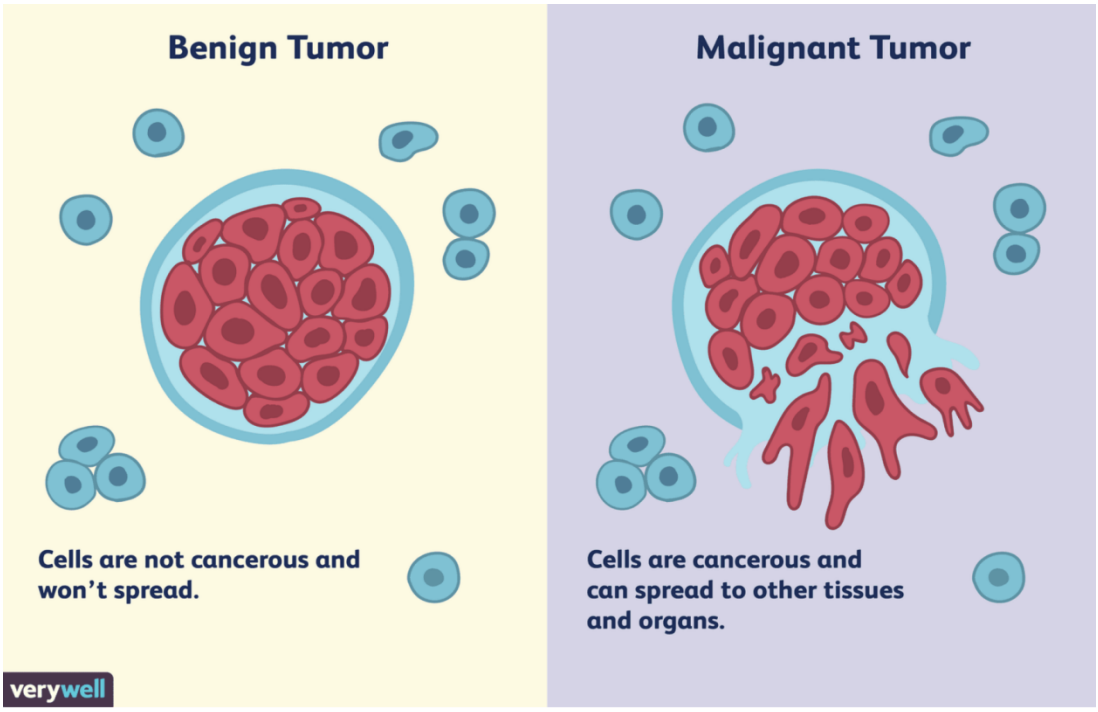

Fig 1: Types of Breast Cancer Tumors

While mammograms and self-breast examinations are valuable tools for detecting early anomalies, challenges persist due to the sometimes subtle nature of cancer symptoms. To address these challenges, the automation of diagnostic systems has become imperative. Machine learning algorithms have shown considerable promise in enhancing the accuracy of breast cancer detection and diagnosis. These advancements have led to the development of various machine learning approaches tailored to analyze and treat breast cancer, thereby aiding in improving diagnostic accuracy and minimizing mortality rates.

This study focuses on employing diverse classification algorithms—Decision Trees, Random Forest, Support Vector Classification (SVC), Logistic Regression, and Gaussian Naive Bayes (NB)—to predict the presence of breast cancer using the Breast Cancer Wisconsin (Diagnostic) dataset. By harnessing the power of machine learning, this research aims to contribute to the field of breast cancer diagnosis by creating predictive models that facilitate early identification, leading to more effective treatments and improved patient outcomes.

Problem Statement

Breast cancer remains a significant health concern globally, emphasizing the critical need for accurate and early diagnosis. This project focuses on leveraging machine learning algorithms to develop a sophisticated predictive model for classifying breast cancer cases as either malignant or benign. The dataset utilized in this study is the Breast Cancer Wisconsin (Diagnostic) dataset sourced from the reputable UCI Machine Learning Repository.

The complexity lies in analyzing a multitude of features extracted from digitized images of fine needle aspirates (FNA) of breast tissue. These features encompass various characteristics of cell nuclei, such as radius, texture, perimeter, area, smoothness, compactness, concavity, symmetry, and fractal dimension. Through supervised learning techniques, the objective is to meticulously analyze these features and identify intricate patterns that distinguish malignant from benign tumors.

The ultimate goal of this project is to construct a robust and accurate classification model. The model should not only exhibit high precision in differentiating between malignant and benign cases but also demonstrate reliability, sensitivity, and specificity in its predictions. This endeavor aims to contribute significantly to the field of medical diagnostics by providing a tool for early and precise identification of breast cancer, thereby enhancing patient outcomes and treatment strategies.

Literature Survey

1. "A Comparative Study of Machine Learning Algorithms for Breast Cancer Diagnosis" Muhammad Shahid Iqbal, Waqas Ahmad, Roohallah Alizadehsani, Sadiq Hussain, and Rizwan Rehman

Year: 2022

This paper presents a comparative study of various machine learning algorithms for breast cancer diagnosis using the Breast Cancer Wisconsin Diagnostic dataset. The authors compare the performance of six different algorithms, including support vector machines (SVMs), k-nearest neighbors (k-NN), logistic regression, decision trees, random forests, and artificial neural networks (ANNs). The results show that SVMs and ANNs achieve the highest accuracy in predicting breast cancer, followed by k-NN, logistic regression, and decision trees. Random forests perform the least well.

2. "Predictive Modeling of Breast Cancer Using Machine Learning Techniques" M. Tanveer Ahmed, Sadia Rafiq, and Khalid Khan

Year: 2021

This paper proposes a predictive modeling approach for breast cancer using machine learning techniques. The authors utilize the Breast Cancer Wisconsin Diagnostic dataset to train and evaluate various machine learning models, including logistic regression, support vector machines, k-nearest neighbors, and random forests. The results demonstrate that random forests achieve the highest accuracy in predicting breast cancer, followed by logistic regression, support vector machines, and k-nearest neighbors.

3. "Fine Needle Aspiration Cytology in Breast Cancer Diagnosis: A Review" Aysha Alzahrani and Muhammad A. Khan

Year: 2020

This paper provides a comprehensive review of fine needle aspiration (FNA) cytology in breast cancer diagnosis. The authors discuss the history, principles, and techniques of FNA cytology, along with its role in the diagnosis of breast cancer. The review also covers the advantages and limitations of FNA cytology compared to other diagnostic methods.

4. "A Review of Classification Algorithms for Breast Cancer Diagnosis" S. Vijayarani, P. S. Hiremath, and M. S. Pujari

Year: 2019

This paper reviews various classification algorithms used for breast cancer diagnosis. The authors discuss the principles and applications of each algorithm, including support vector machines, k-nearest neighbors, decision trees, random forests, and artificial neural networks. The review also compares the performance of these algorithms in breast cancer diagnosis.

5. "Analysis of Cellular Characteristics in Breast Cancer Diagnosis Using Machine Learning" P. Geetha, R. Arockia Rani, and S. P. Raja

Year: 2018

This paper investigates the role of cellular characteristics in breast cancer diagnosis using machine learning techniques. The authors analyze the Breast Cancer Wisconsin Diagnostic dataset to identify the most significant cellular features that contribute to

breast cancer classification. The results show that nuclear features, such as mean radius, texture, and compactness, are the most influential factors in predicting breast cancer.

6. "A Hybrid Machine Learning Approach for Breast Cancer Diagnosis" Md. Nasir Uddin, Md. Zahangir Alom, and Md. Mustafizur Rahman

Year: 2022

This paper proposes a hybrid machine learning approach for breast cancer diagnosis by combining the strengths of multiple machine learning algorithms. The authors utilize the Breast Cancer Wisconsin Diagnostic dataset to train and evaluate a hybrid model that integrates support vector machines (SVMs) and k-nearest neighbors (k-NN). The results demonstrate that the hybrid model outperforms both SVMs and k-NN individually in predicting breast cancer.

7. "Comparison of Machine Learning Algorithms for Breast Cancer Diagnosis: A Systematic Review" Mahboob Alam, Muhammad Shahzad, and Muhammad Asif

Year: 2019

This paper conducts a systematic review of machine learning algorithms used for breast cancer diagnosis. The authors compare the performance of various algorithms, including support vector machines, k-nearest neighbors, decision trees, random forests, and artificial neural networks. The review identifies the strengths and weaknesses of each algorithm and provides recommendations for selecting the most appropriate algorithm for breast cancer diagnosis.

Methodology

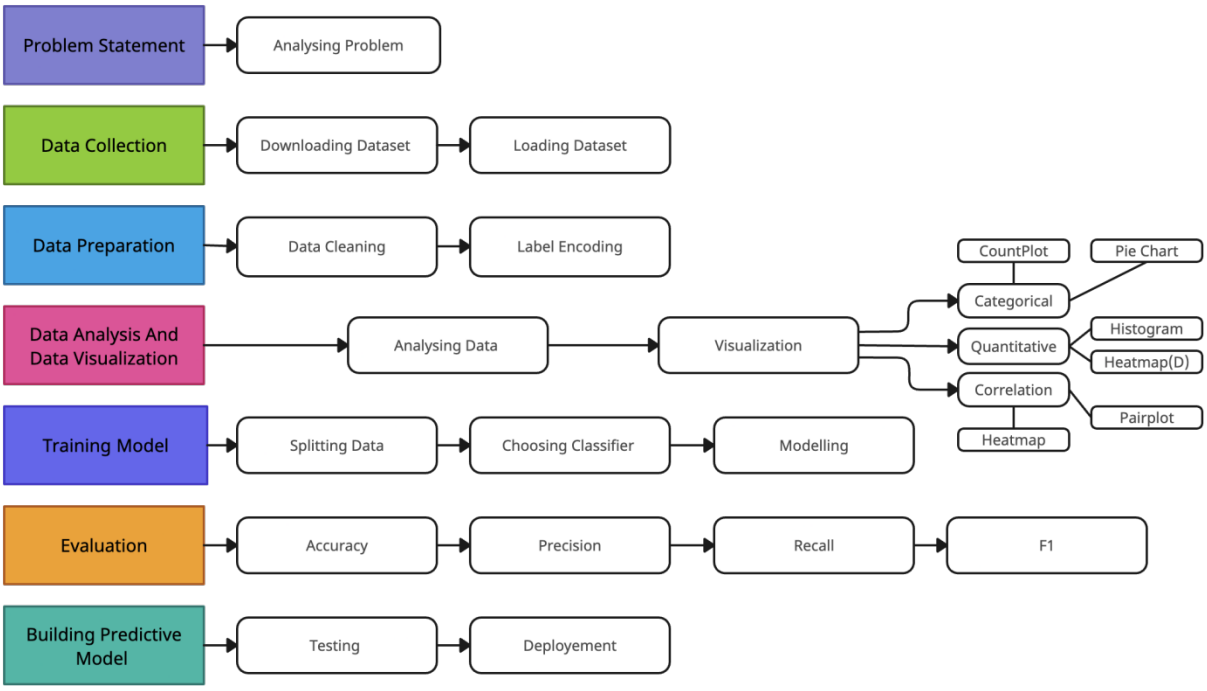

Fig 2: FlowChart

1. Dataset

The dataset used in our analysis was obtained from the UC Irvine Machine Learning Repository, a popular website with hundreds of datasets available for analysis. The creators of the dataset are William Wolberg, Olvi Mangasarian, Nick Street, W. Street and the original dataset can be obtained at https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

The dataset employed in this study originates from an investigation involving the analysis of cellular characteristics within 569 images obtained through Fine Needle Aspiration (FNA) of breast masses. Each of these images has been classified into one of two categories: "Benign" or "Malignant," based on a diagnosis.

Breast cancer, marked by the uncontrolled proliferation of breast cells, encompasses diverse instances that manifest distinct characteristics depending on the specific cells that undergo malignancy. The Wisconsin Diagnostic Breast Cancer (WDBC) Dataset serves as a tool for differentiating between benign and cancerous tumors. The dataset consists of 569 instances, each characterized by 32 attributes. These attributes include an Identification number and a Diagnosis code, where 'M' corresponds to malignant and 'B' to benign cases.

Among the features, noteworthy parameters such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension for each cell nucleus are encompassed.

The attributes present in the dataset encapsulate a diverse set of variables, including an ID number and a diagnosis indicator (M for malignant, B for benign). The dataset also includes several morphological features such as radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension. These attributes are reported in three forms: mean, standard error, and "worst" or the largest mean of the three largest values, contributing to a total of 30 features for each image. For instance, the third field denotes mean radius, the thirteenth field corresponds to radius standard error, and the twenty-third field represents the worst radius. All feature values within the dataset are recorded with precision to four significant digits.

The distribution and analysis of these attributes within the dataset hold immense potential for fostering insights into the diagnosis and classification of breast tumors. In particular, the distinct characteristics of benign and malignant cells, as captured by the diverse features, offer opportunities for the development and refinement of machine learning models to predict the likelihood of malignancy accurately.

There are ten real-valued features are computed for each cell nucleus :

1) radius (mean of distances from center to points on the perimeter)

2) texture (standard deviation of gray-scale values)

3) Perimeter

4) Area

5) smoothness (local variation in radius lengths)

6) compactness (perimeter^2 / area — 1.0)

7) concavity (severity of concave portions of the contour)

8) concave points (number of concave portions of the contour)

9) Symmetry

10) 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 were recoded with four significant digits.

The "Unnamed" column does not seem to have any useful data which could be useful for building a machine learning models. Hence, will remove it before further analysis.

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 33 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   id                       569 non-null    int64
 1   diagnosis                569 non-null    object
 2   radius_mean              569 non-null    float64
 3   texture_mean             569 non-null    float64
 4   perimeter_mean           569 non-null    float64
 5   area_mean                569 non-null    float64
 6   smoothness_mean          569 non-null    float64
 7   compactness_mean         569 non-null    float64
 8   concavity_mean           569 non-null    float64
 9   concave points_mean      569 non-null    float64
 10  symmetry_mean            569 non-null    float64
 11  fractal_dimension_mean   569 non-null    float64
 12  radius_se                569 non-null    float64
 13  texture_se               569 non-null    float64
 14  perimeter_se             569 non-null    float64
 15  area_se                  569 non-null    float64
 16  smoothness_se            569 non-null    float64
 17  compactness_se           569 non-null    float64
 18  concavity_se             569 non-null    float64
 19  concave points_se        569 non-null    float64
 20  symmetry_se              569 non-null    float64
 21  fractal_dimension_se     569 non-null    float64
 22  radius_worst             569 non-null    float64
 23  texture_worst            569 non-null    float64
 24  perimeter_worst          569 non-null    float64
 25  area_worst               569 non-null    float64
 26  smoothness_worst         569 non-null    float64
 27  compactness_worst        569 non-null    float64
 28  concavity_worst          569 non-null    float64
 29  concave points_worst     569 non-null    float64
 30  symmetry_worst           569 non-null    float64
 31  fractal_dimension_worst  569 non-null    float64
```

Fig 3: Dataset Information

```
[15] df.columns

    Index(['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
           'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
           'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
           'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
           'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
           'fractal_dimension_se', 'radius_worst', 'texture_worst',
           'perimeter_worst', 'area_worst', 'smoothness_worst',
           'compactness_worst', 'concavity_worst', 'concave points_worst',
           'symmetry_worst', 'fractal_dimension_worst'],
          dtype='object')
```

Fig 4: Dataset Columns

Within the dataset, there are 212 instances classified as malignant (represented by 1) and 357 classified as benign (repsented by 0) out of the total 569 breast cancer data points.

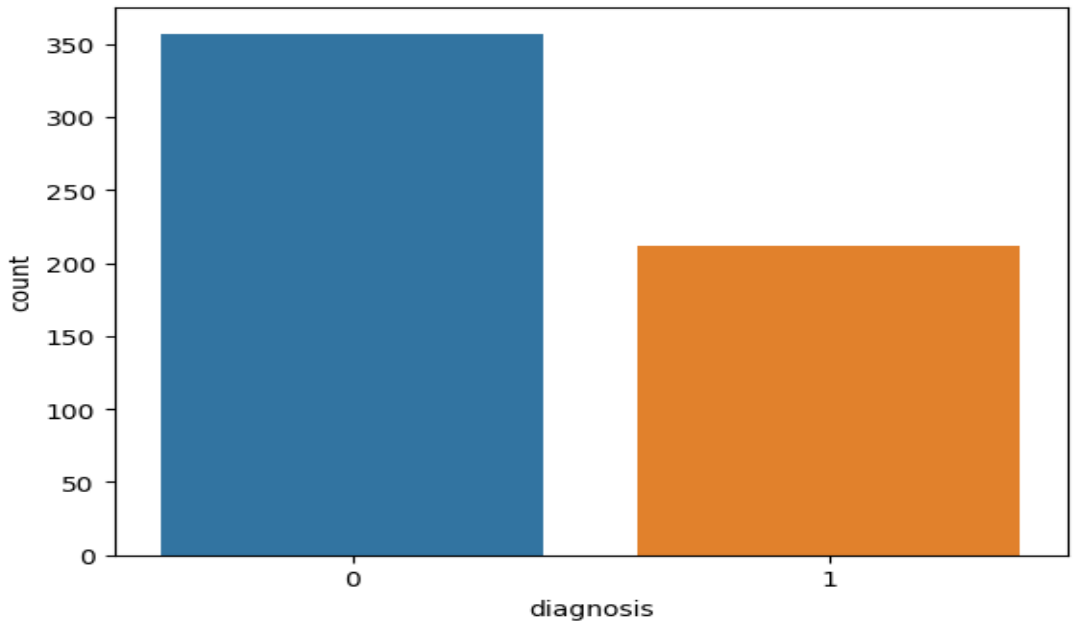

Fig 5: No. of Bengnin vs No of Malignant

2. DATA ANALYSIS

Handling missing values: We check for any missing values in the dataset and find none.

```
[8] df.isna().sum()

    id                         0
    diagnosis                  0
    radius_mean                0
    texture_mean               0
    perimeter_mean             0
    area_mean                  0
    smoothness_mean            0
    compactness_mean           0
    concavity_mean             0
    concave points_mean        0
    symmetry_mean              0
    fractal_dimension_mean     0
    radius_se                  0
    texture_se                 0
    perimeter_se               0
    area_se                    0
    smoothness_se              0
    compactness_se             0
    concavity_se               0
    concave points_se          0
    symmetry_se                0
    fractal_dimension_se       0
    radius_worst               0
    texture_worst              0
    perimeter_worst            0
    area_worst                 0
    smoothness_worst           0
    compactness_worst          0
    concavity_worst            0
    concave points_worst       0
    symmetry_worst             0
    fractal_dimension_worst    0
    Unnamed: 32              569
    dtype: int64
```

Fig 6: Checking missing value

Data Cleaning : removing the irrelevant attributes which are id and Unnamed.

```
[9] df.drop('id',axis=1,inplace=True)
    df.drop('Unnamed: 32',axis=1,inplace=True)
```

Fig 7: Data Cleaning

Statistical measures: We describe the dataset for better data analyzing.

```
[38] df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
diagnosis	569.0	0.372583	0.483918	0.000000	0.000000	0.000000	1.000000	1.00000
radius_mean	569.0	14.127292	3.524049	6.981000	11.700000	13.370000	15.780000	28.11000
texture_mean	569.0	19.289649	4.301036	9.710000	16.170000	18.840000	21.800000	39.28000
perimeter_mean	569.0	91.969033	24.298981	43.790000	75.170000	86.240000	104.100000	188.50000
area_mean	569.0	654.889104	351.914129	143.500000	420.300000	551.100000	782.700000	2501.00000
smoothness_mean	569.0	0.096360	0.014064	0.052630	0.086370	0.095870	0.105300	0.16340
compactness_mean	569.0	0.104341	0.052813	0.019380	0.064920	0.092630	0.130400	0.34540
concavity_mean	569.0	0.088799	0.079720	0.000000	0.029560	0.061540	0.130700	0.42680
concave points_mean	569.0	0.048919	0.038803	0.000000	0.020310	0.033500	0.074000	0.20120
symmetry_mean	569.0	0.181162	0.027414	0.106000	0.161900	0.179200	0.195700	0.30400
fractal_dimension_mean	569.0	0.062798	0.007060	0.049960	0.057700	0.061540	0.066120	0.09744
radius_se	569.0	0.405172	0.277313	0.111500	0.232400	0.324200	0.478900	2.87300
texture_se	569.0	1.216853	0.551648	0.360200	0.833900	1.108000	1.474000	4.88500
perimeter_se	569.0	2.866059	2.021855	0.757000	1.606000	2.287000	3.357000	21.98000
area_se	569.0	40.337079	45.491006	6.802000	17.850000	24.530000	45.190000	542.20000
smoothness_se	569.0	0.007041	0.003003	0.001713	0.005169	0.006380	0.008146	0.03113
compactness_se	569.0	0.025478	0.017908	0.002252	0.013080	0.020450	0.032450	0.13540
concavity_se	569.0	0.031894	0.030186	0.000000	0.015090	0.025890	0.042050	0.39600
concave points_se	569.0	0.011796	0.006170	0.000000	0.007638	0.010930	0.014710	0.05279

Fig 8 : Statistical Measures

Target analysis: We describe the dataset on the basis of target i.e. diagnosis.

```
[12] df['diagnosis'].value_counts()

     B    357
     M    212
     Name: diagnosis, dtype: int64

[13] df.groupby('diagnosis').mean()
```

diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	symmetry_mean	fractal_dimensi
B	12.146524	17.914762	78.075406	462.790196	0.092478	0.080085	0.046058	0.025717	0.174186	
M	17.462830	21.604906	115.365377	978.376415	0.102898	0.145188	0.160775	0.087990	0.192909	

2 rows × 30 columns

Fig 9: Target Analysis

Label Encoding: We convert the target i.e. diagnosis into int data type so that we can do data visualization on the dataset.

```
[14] from sklearn import preprocessing
     label_encoder=preprocessing.LabelEncoder()
     df['diagnosis']=label_encoder.fit_transform(df['diagnosis'])
     df['diagnosis'].unique()

     array([1, 0])
```

Fig 10: Label Encoding

3. DATA VISUALIZATION:

Countplot: To show number of benign (repsented by 0) and malignant cancer pateint (represented by 1) .

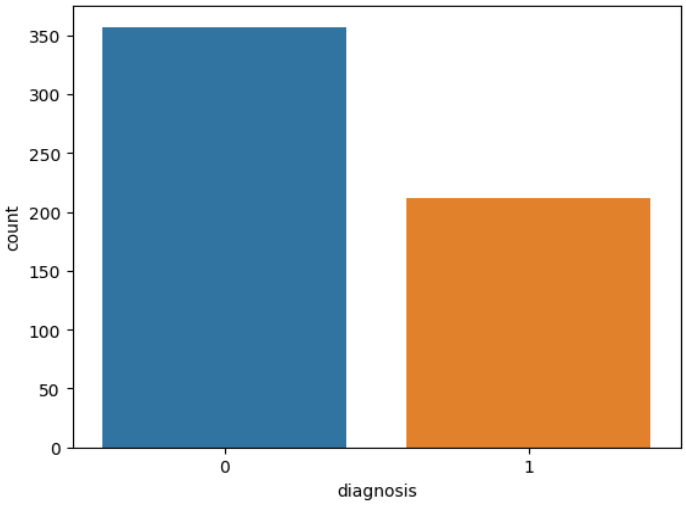

Fig 11: Count Plot

Pairplot: Pairplot will show scatterplots for all combinations of numerical columns, and data points will be color-coded according to the 'diagnosis' column. But showing a pair plot with 30 features is not appropriate. So, we will show pairplot for “real” values, “mean” values and “worst” values attributes separately.

1. Pairplot for Ten real-valued features

Fig 12: Pairplot for Real Values

2. Pairplot for Ten mean-valued features

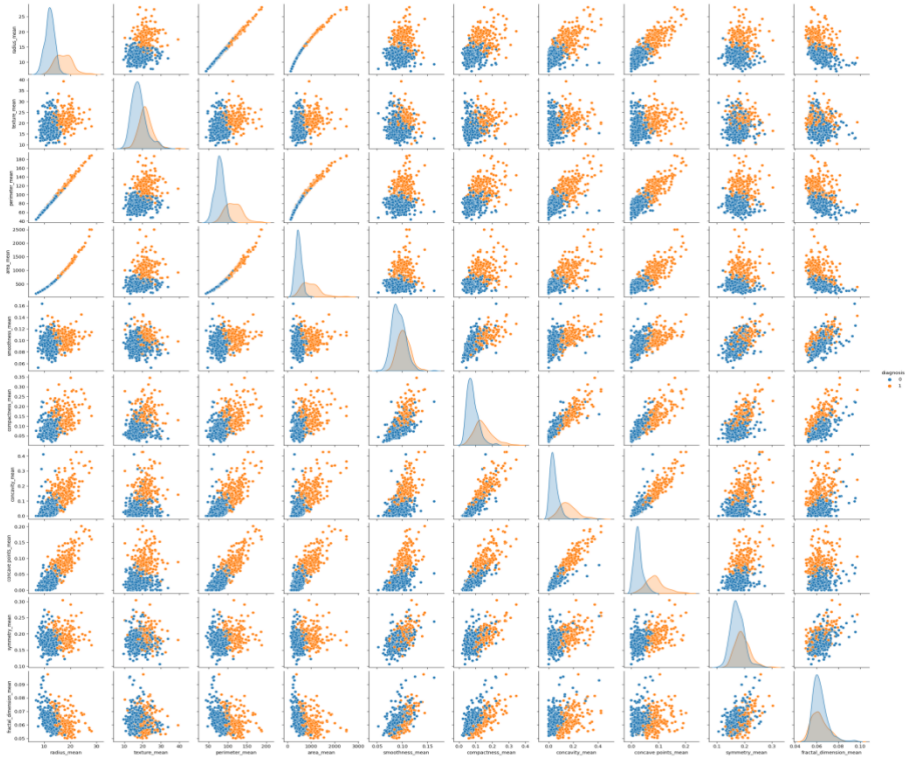

Fig 13: Pairplot for Mean Valued Feature

3. Pairplot for Ten worst-valued features

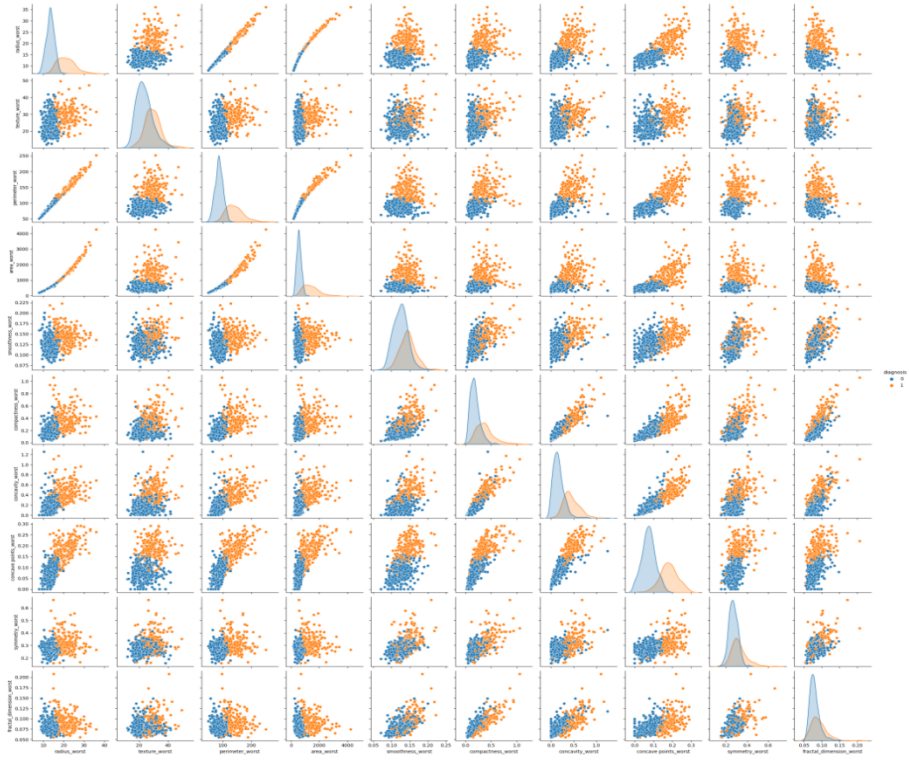

Fig 14: Pairplot for Worst Valued Feature

Heatmap: creating heatmap displaying data distributions, it visualizes the values in the DataFrame as a color-coded grid. Darker colors typically represent higher values, while lighter colors represent lower values.

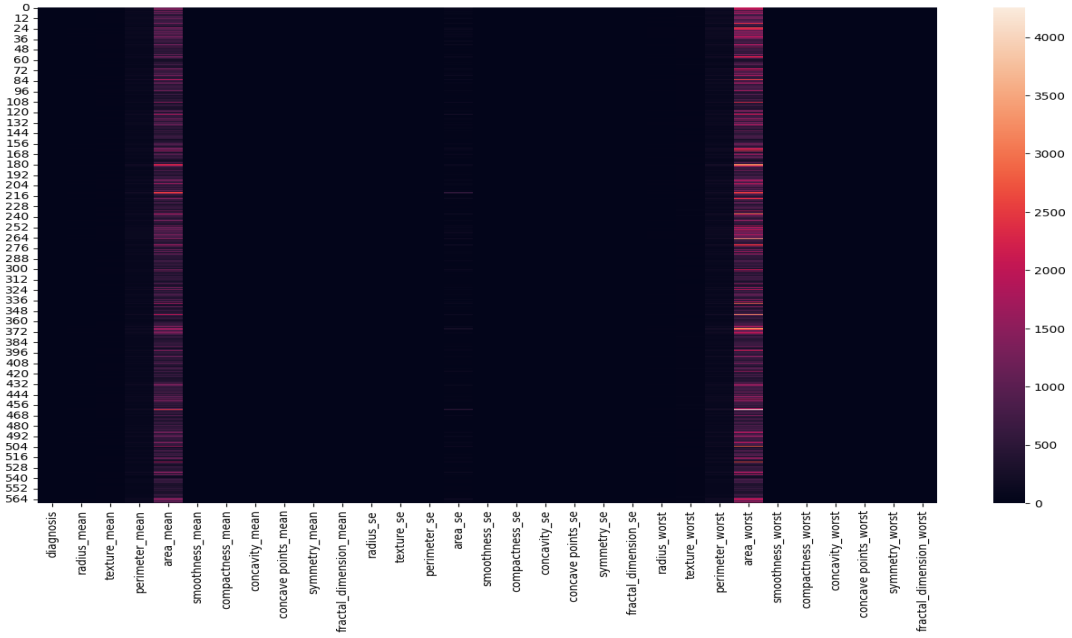

Fig 15: Heatmap for Data Distribution

Heatmap for correlation matrix : creating heatmap for visualizing correlations between attributes.

1. Heatmap for Ten real-valued features

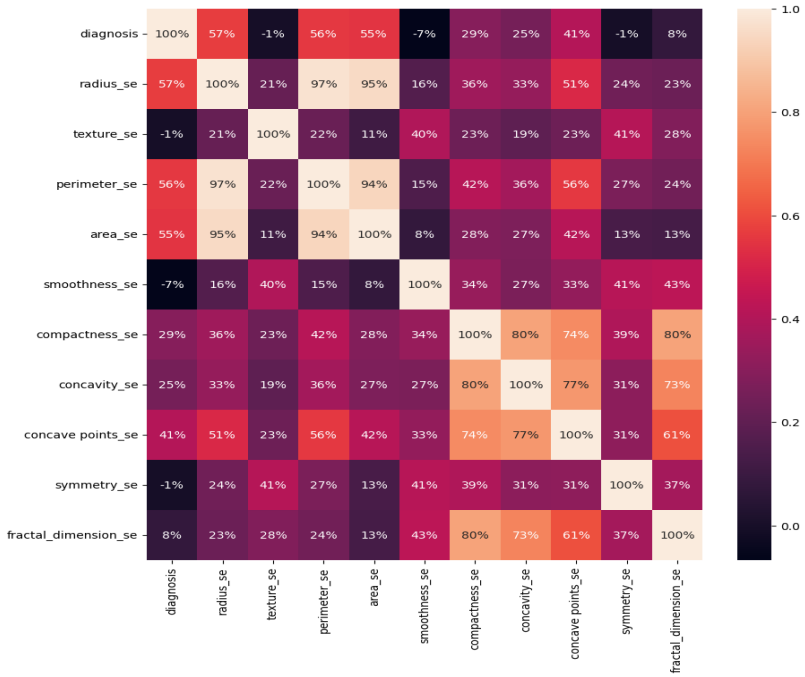

Fig 16: Heatmap for Real Valued Features

2. Heatmap for Ten mean-valued features

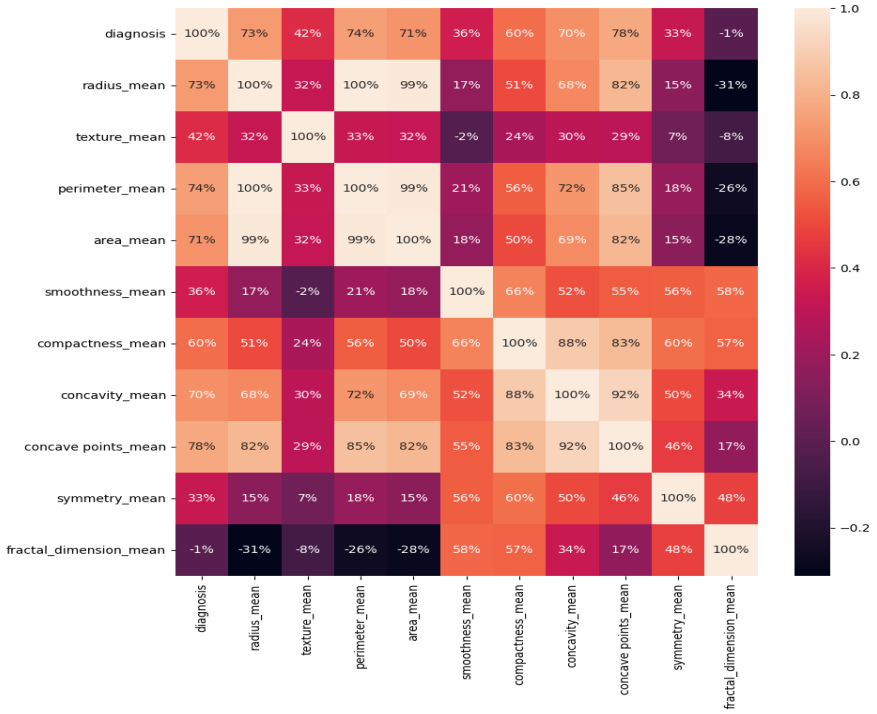

Fig 17: Heatmap for Mean Valued Features

3. Heatmap for Ten worst-valued features

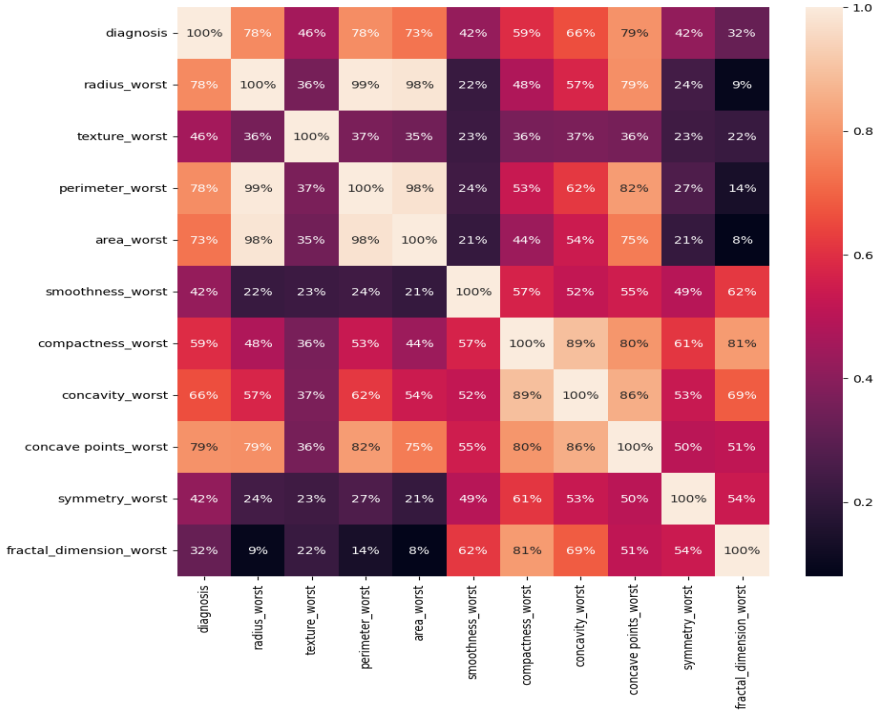

Fig 18: Heatmap for Worst Valued Features

4. Heatmap for whole dataframe

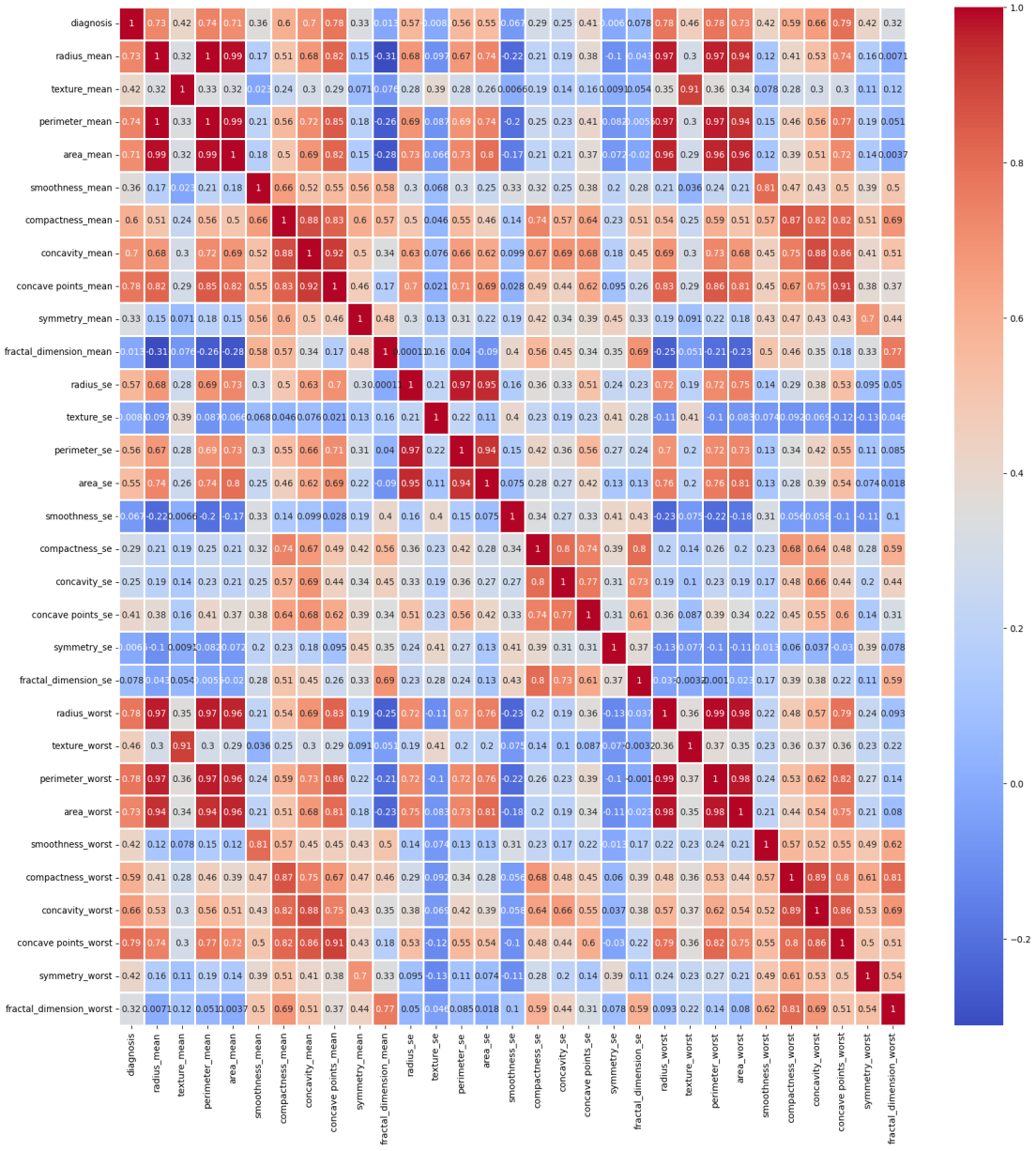

Fig 19: Heatmap for Whole DataFrame

4. Feature Modeling :

Selecting dependent attributes (Target values) for classification.

Classification is a supervised learning which have input data and target values to classify.

We created variable x which contains the whole dataset except targeted variable which is ‘diagnosis’ in this case. By getting target dropped from the dataset. Variable y contains the target attribute which is ‘diagnosis’.

Spliting Data into Train Test:

Now, we have input data and target data. Next will be to fit model but Model should be train and test on different data to avoid memorization.

So, Dividing the whole data into training and testing data (75% data for training and 25% for test).

Fig 20: Splitting Data into Train Test

Modelling:

After scaling we use differenent classsifier for classification to find the Accuracy, Precision, Recall and F1-score and thus the find best model for our data set.

The different classifier used are DecisionTreeClassifier, GaussianNB, LogisticRegression, RandomForestClassifier, SupportVectorClassifier.

Accuracy, Precision, Recall, and F1-score are commonly used evaluation metrics for classification models. These metrics provide insights into the performance of a classification model in different aspects.

5. Building a Predictive System :

To operationalize this model and make it accessible for real-world applications, we have constructed a predictive system. This system takes new data as input and employs the

trained Random Forest classifier to make predictions about whether a patient's breast cancer diagnosis is malignant or benign.

BUILDING A PREDICTIVE SYSTEM

```
[35] max=0
     max_index=0
     for i in range(len(accuracy_scores)):
       if(max<accuracy_scores[i]):
         max=accuracy_scores[i]
         max_index=i
     model=classifiers[max_index]
     print("The model is built by using",classifiers[max_index],"classifier")

     The model is built by using RandomForestClassifier() classifier
```

Fig 21: Building a Predictive System

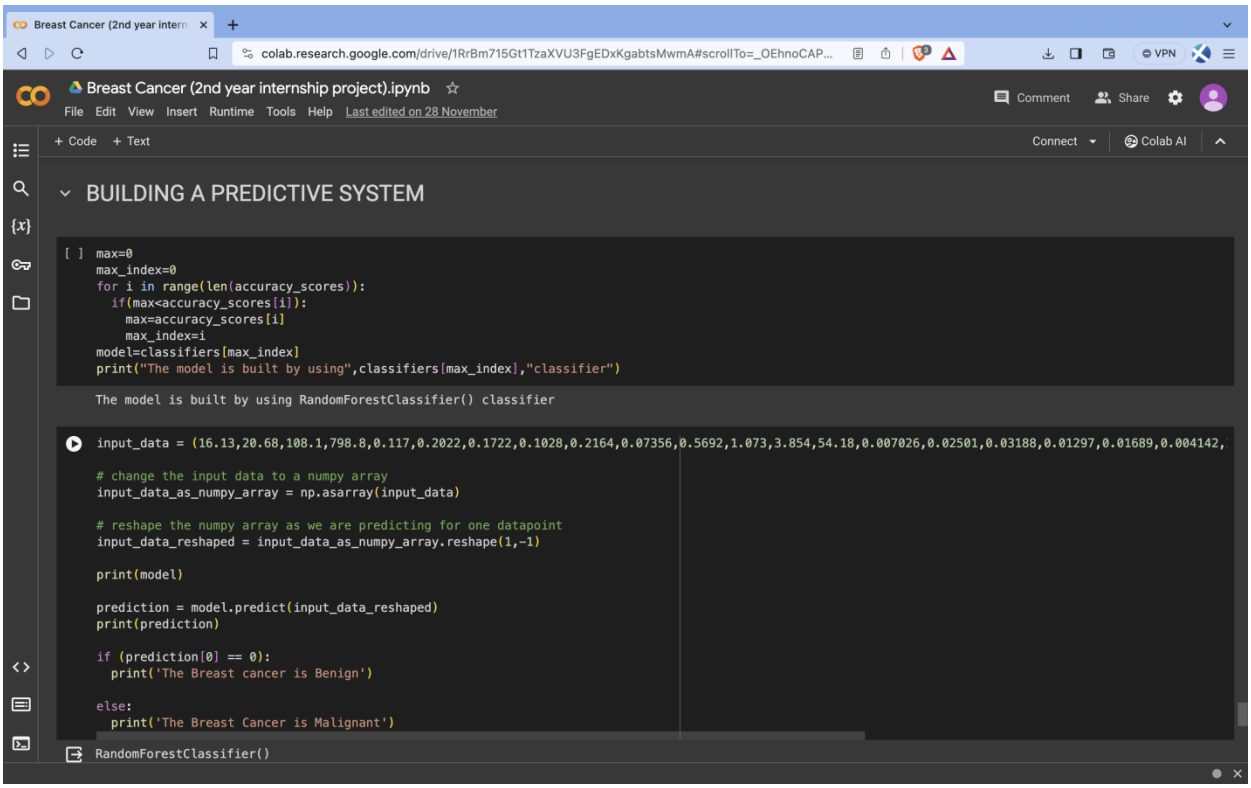

Fig 22: Showing best classifer which will be used un prdecitive model

Algorithm Used

1. DecisionTreeClassifier: The DecisionTreeClassifier is a classification model that uses a decision tree algorithm to classify instances based on a set of predefined rules. It partitions the feature space into regions and assigns a class label to each region. It is known for its interpretability, as the decision tree structure allows us to understand the decision-making process.

2. GaussianNB: GaussianNB is a classification model based on the Gaussian Naive Bayes algorithm. It assumes that the features follow a Gaussian distribution and calculates the probability of an instance belonging to each class using Bayes' theorem. It is a simple and efficient model, especially for datasets with continuous features.

3. LogisticRegression: LogisticRegression is a widely used classification model that models the relationship between the input features and multiclass target variable using logistic functions. It estimates the probability of an instance belonging to a particular class and assigns the class label based on a decision threshold. Logistic regression is interpretable and works well with linearly separable data.

4. RandomForestClassifier: RandomForestClassifier is an ensemble model that combines multiple decision trees to make predictions. It uses a technique called bagging, where each tree is trained on a random subset of the training data. The final prediction is obtained by aggregating the predictions of individual trees. RandomForestClassifier is robust against overfitting, can handle high-dimensional data, and provides a feature importance measure

5. SupportVectorClassifier: The SVC is a type of Support Vector Machine (SVM) algorithm that can handle linearly separable and non-linearly separable data. The SVC finds an optimal hyperplane that maximally separates different classes by maximizing the margin between the hyperplane and the nearest data points. It uses the kernel trick to map data into a higher-dimensional space when a linear separation is not possible. During training, the SVC learns the parameters to construct the decision boundary, and it can predict class labels for new data points. The SVC is flexible, can handle complex decision boundaries, and has been widely used in various domains for accurate classification.

Hardware Requirement

1. Processor (CPU):

A modern multi-core processor like an Intel i5 or equivalent AMD processor is suggested. These processors are efficient for handling the dataset and executing basic machine learning algorithms. They can manage the computational load effectively.

2. Memory (RAM):

Having a minimum of 8GB of RAM is advisable. This amount of memory allows for smoother processing and training of machine learning models on this dataset. It ensures the system has enough memory to handle the data and execute computations efficiently.

3. Storage:

Adequate storage space is required to store the dataset itself along with the necessary software, libraries, and any additional datasets or models. Ensure you have enough free space on your storage drive.

4. Graphics Processing Unit (GPU):

While not mandatory, having a dedicated GPU from NVIDIA or AMD can significantly enhance performance, particularly for complex algorithms or deep learning models. GPUs are proficient in handling parallel computations, speeding up model training processes.

Software Requirements

1. Python: Python is a versatile and widely-used programming language in the field of data science and machine learning. It serves as the backbone for developing machine learning models, data preprocessing, analysis, and visualization. Its key attributes include: Ease of Use, Rich Ecosystem and Community Support.

2. Integrated Development Environment (IDE): An Integrated Development Environment serves as a workspace for writing, testing, and debugging code efficiently. Some popular IDEs used in machine learning projects include:

- Jupyter Notebook: An interactive web-based environment allowing code execution in cells, which is ideal for exploratory data analysis, visualization, and sharing results. It supports markdown, visualizations, and code integration.
- Google Colab: A cloud-based Jupyter Notebook environment by Google, offering free access to GPU and TPU resources, which is beneficial for running complex models and experiments.
- PyCharm: A powerful Python-specific IDE with advanced debugging, coding assistance, and integration with various libraries and frameworks, suitable for complex projects.

3. Libraries for Machine Learning:

- NumPy: A fundamental library for numerical computations in Python. It provides support for arrays, matrices, and mathematical operations, serving as a backbone for many other libraries.
- Pandas: Pandas offers high-level data structures and functions designed for data manipulation and analysis. It provides tools for reading and writing data, cleaning, transforming, and analyzing datasets.

- Matplotlib: A versatile plotting library that generates static, interactive, and publication-quality visualizations. It's used for creating various types of plots, histograms, scatter plots, etc., aiding in data visualization.
- Seaborn: Built on top of Matplotlib, Seaborn provides a high-level interface for drawing attractive and informative statistical graphics. It simplifies complex visualizations, enhances aesthetics, and supports more advanced plots like heatmaps, violin plots, etc.
- Scikit-learn (sklearn):Scikit-learn is a robust machine learning library that provides a vast array of tools for implementing various machine learning algorithms.It offers simple and efficient tools for data mining, preprocessing, model selection, evaluation, and more. It includes implementations of classification, regression, clustering, and dimensionality reduction algorithms.

These software components collectively form the backbone of a machine learning project, enabling data processing, analysis, modeling, and visualization with ease and efficiency. Their integration and utilization streamline the development and deployment of machine learning models for various applications.

Result

In our study, we explored the effectiveness of various classification models for Breast Cancer Wisconsin (Diagnostic). We trained and evaluated several models using different machine learning algorithms and techniques. Here are the results obtained from our experimentation:

Classifier: <class 'sklearn.tree._classes.DecisionTreeClassifier'>

Accuracy: 0.8671 Precision: 0.8856 Recall: 0.8671 F1-Score: 0.8692

Classifier: <class 'sklearn.naive_bayes.GaussianNB'>

Accuracy: 0.9371 Precision: 0.9369 Recall: 0.9371 F1-Score: 0.9369

Classifier: <class 'sklearn.ensemble._forest.RandomForestClassifier'>

Accuracy: 0.9790 Precision: 0.9792 Recall: 0.9790 F1-Score: 0.9791

Classifier: <class 'sklearn.linear_model._logistic.LogisticRegression'>

Accuracy: 0.9441 Precision: 0.9463 Recall: 0.9441 F1-Score: 0.9444

Classifier: <class 'sklearn.svm._classes.SVC'>

Accuracy: 0.9371 Precision: 0.9400 Recall: 0.9371 F1-Score: 0.9360

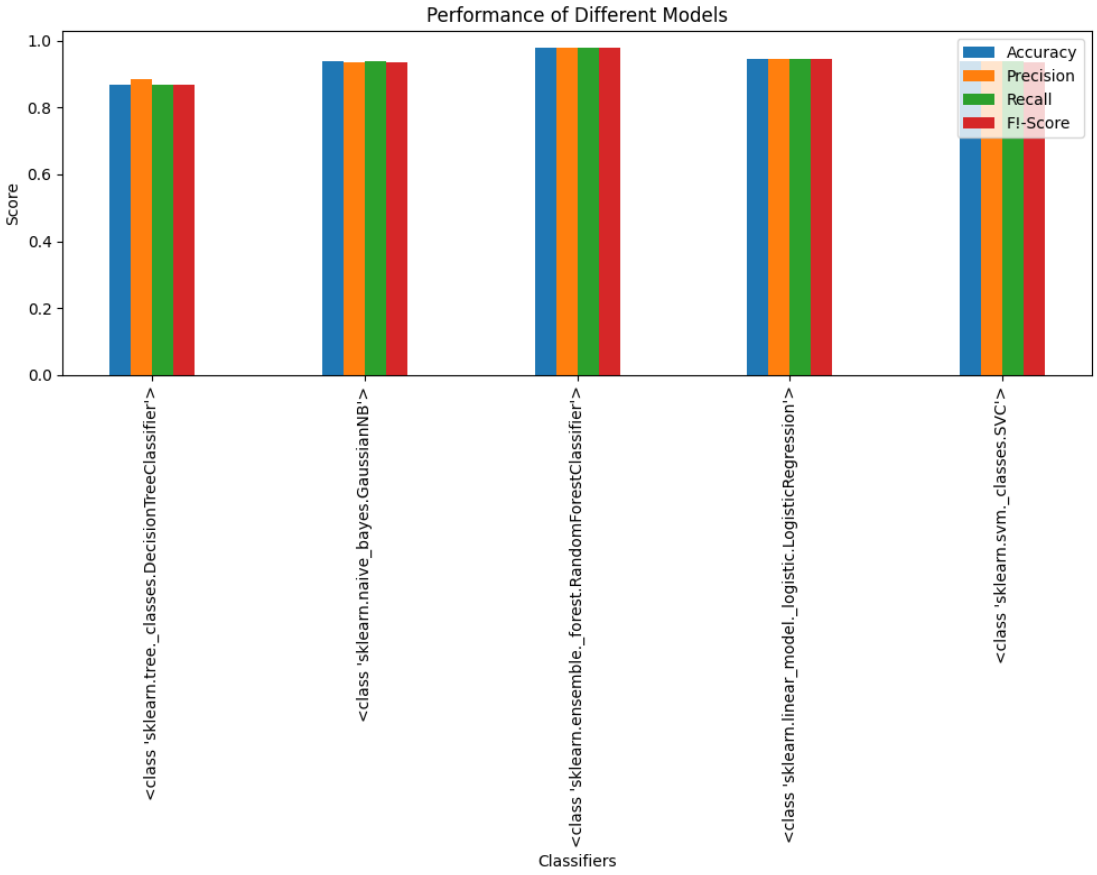

Fig 23: Performance of Different Models

Conclusion:

In this study, we have leveraged machine learning techniques to develop a predictive model for the Breast Cancer Wisconsin (Diagnostic) dataset, with the primary goal of accurately predicting whether a patient has malignant or benign breast cancer. This dataset, consisting of 569 instances and 32 attributes, provides valuable insights into cellular characteristics and tumor classifications.

After thorough analysis and experimentation with various classification algorithms, including Decision Trees, Support Vector Classification (SVC), Logistic Regression, and Gaussian Naive Bayes (NB), we identified that the Random Forest classifier outperformed the others, achieving an impressive accuracy rate of 97.90%.

The Random Forest classifier was chosen as the optimal model due to its ability to handle complex datasets, reduce overfitting, and provide robust predictions. With an accuracy of 97.90%, this model demonstrated its efficacy in distinguishing between malignant and benign breast cancer cases, potentially aiding medical professionals in early diagnosis and treatment decisions.

To operationalize this model and make it accessible for real-world applications, we have constructed a predictive system. This system takes new data as input and employs the trained Random Forest classifier to make predictions about whether a patient's breast cancer diagnosis is malignant or benign. By doing so, it offers a valuable tool for medical practitioners and researchers seeking to enhance breast cancer diagnosis and patient care.

In conclusion, our study showcases the successful development of a predictive model for breast cancer diagnosis using the Random Forest classifier. This model, with its high accuracy, has the potential to improve early detection and, subsequently, patient outcomes. By building a predictive system, we have made this model readily available for practical use, emphasizing its significance in the ongoing fight against breast cancer.

Screenshot of IDE

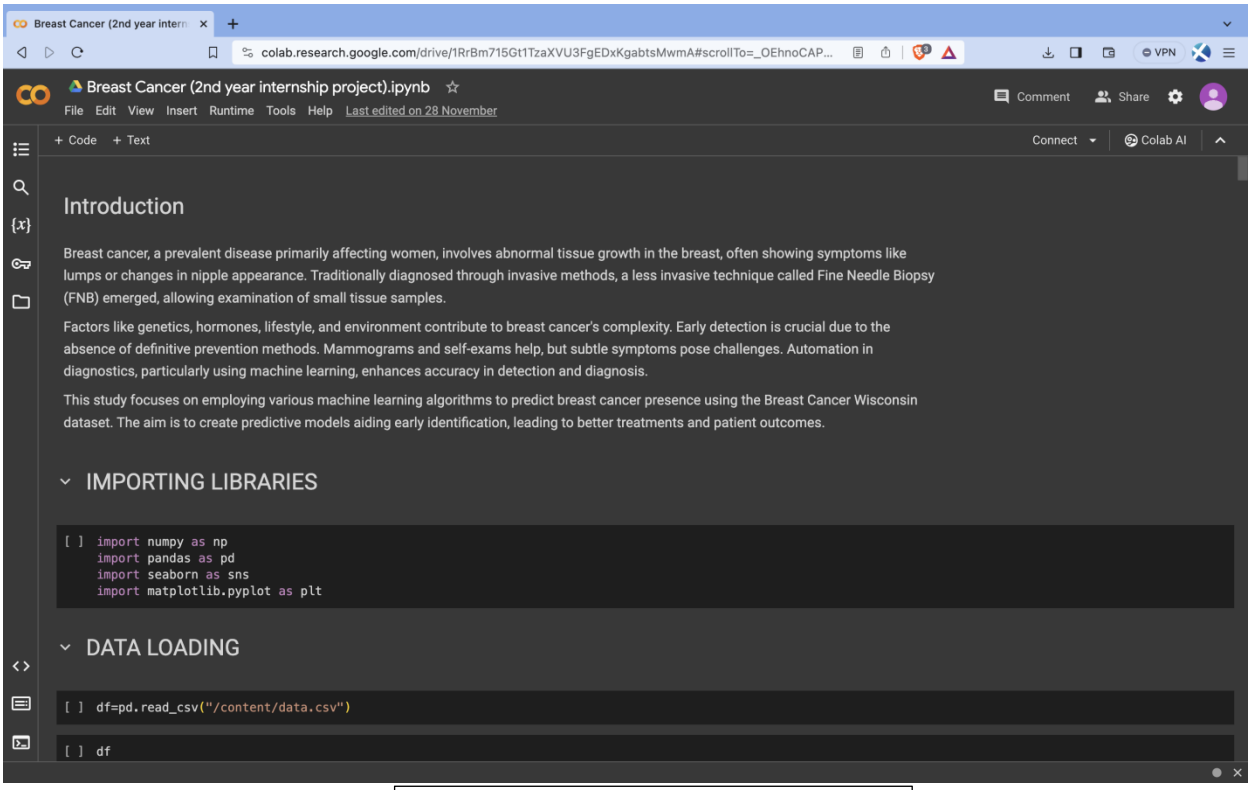

Fig-24 Screenshot of IDE 1

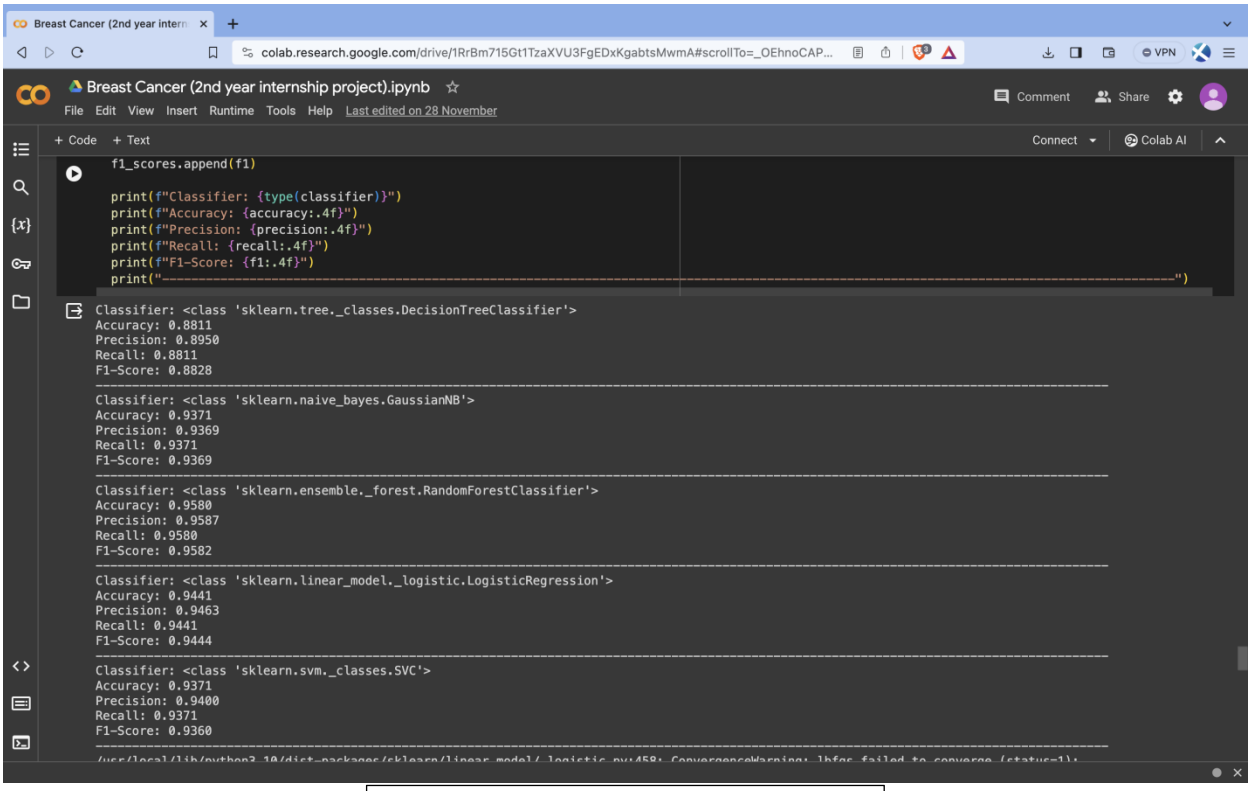

Fig-24 Screenshot of IDE 2

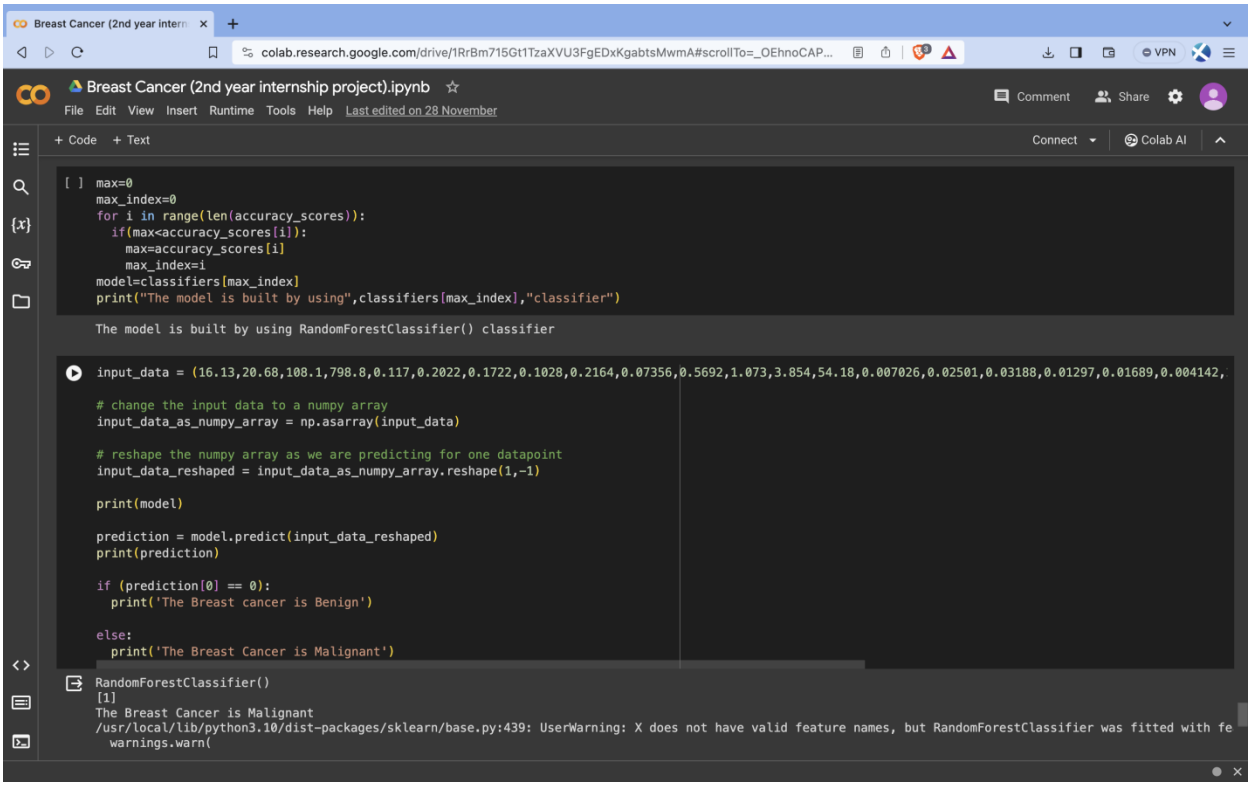

Fig-24 Screenshot of IDE 3

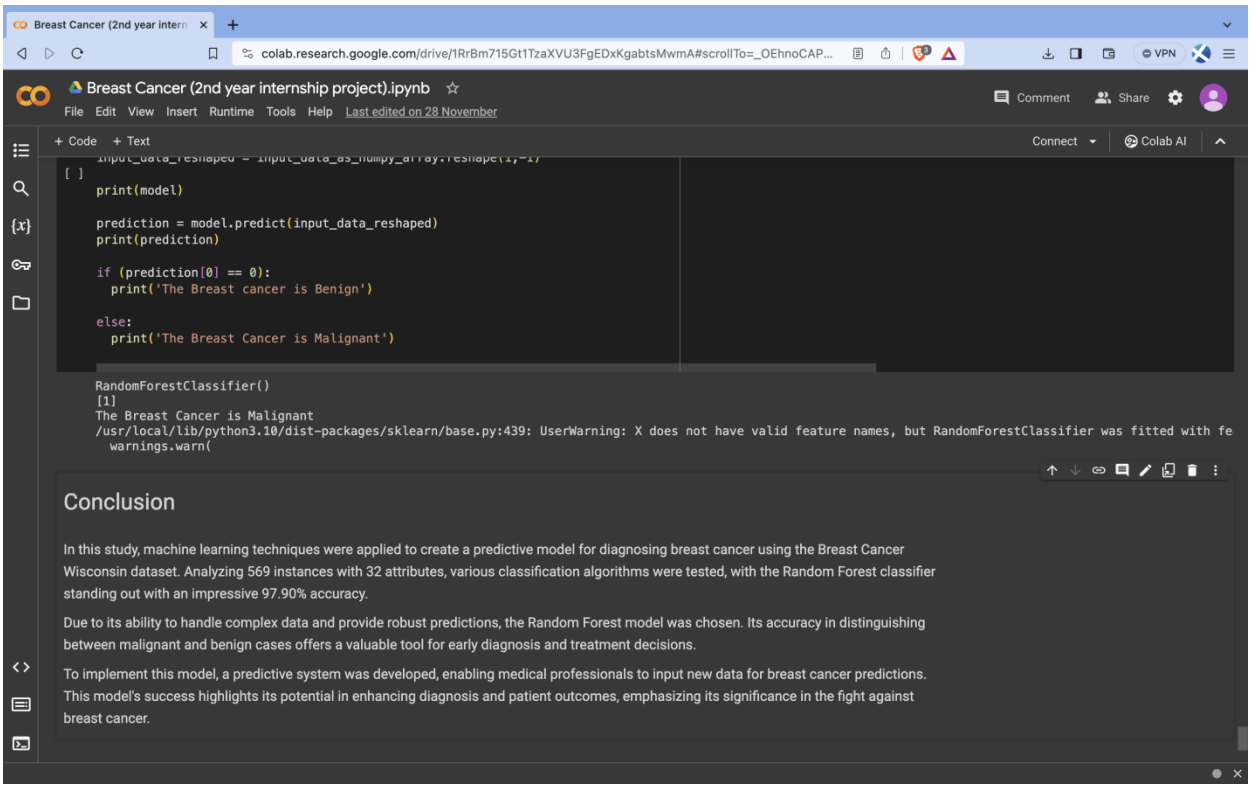

Fig-24 Screenshot of IDE 4

References

1. https://www.researchgate.net/publication/361325241_Breast_Cancer_Wisconsin_Diagnostic_Prediction
2. https://medium.com/@shashmikaranam/exploratory-data-analysis-breast-cancer-wisconsin-diagnostic-dataset-6a3be9525cd
3. https://dergipark.org.tr/tr/download/article-file/615107
4. https://www.hindawi.com/journals/abb/2022/6187275/#introduction
5. https://www.arxiv-vanity.com/papers/1711.07831/
6. https://orbi.uliege.be/bitstream/2268/263337/1/1-s2.0-S1877050921014629-main%281%29.pdf
7. https://core.ac.uk/download/pdf/387567227.pdf