

UNIVERSITY

ABDELMALEK ESSAADI

National School of applied

Sciences of Al Hoceima



END of Semester Project

Field : Data Engineering

Breast Cancer Tumor Classification (Malignant or Benign)

Realized By:

- TATI Mohammed
- JALILI Badr Eddine

Supervisor:

- ROUTAIB Hayat

DEFENDED ON 12-DEC-22, IN THE PRESENCE OF:

- Pr. ROUTAIB Hayat

Acknowledgements

*At the beginning of the year, we needed guidance, sharp remarks, encouragement and helpful instructions, thankfully **Pr. Routaib Hayat** provided us all of that and more with no hesitation , and for that we are forever in her debt, we know this little acknowledgement could not return her favor, Although we are sincerely grateful and blessed to have such a brilliant supervisor.*

Also, it will be ridiculous to not thank her for giving us the opportunity to deploy our skills into a real world case project.

Last but not least, we would not forget to thank anyone who spends time and effort reading this report. we hope it is informative and insightful to you.

Table of Content

Acknowledgements	2
Table of Content	3
Table of Figures	5
List of Abbreviations	6
Problem Statement	7
CHAPTER 1: Exploratory Data Analysis (EDA)	8
1. Work Environment	8
1.1 Languages	8
Python	8
HTML	8
CSS	9
JavaScript	9
1.2 Packages	10
Scikit-learn	10
Pandas	10
Matplotlib	10
Seaborn	11
NumPy	11
JQuery	11
1.3 Frameworks	12
Flask	12
2. Data description	12
2.1 About Dataset:	12
2.2 Features:	13
2.3 Descriptive statistics	15
2.3.1 Data dimension:	15
2.3.2 Data Information:	15
2.4 Data Visualization:	16
2.4.1 Distribution of the malignancy in the dataset:	16
2.4.2 Distribution of data via Histograms grouped by malignancy:	17
2.4.3 Distribution of data via BoxPlots grouped by malignancy:	18
2.4.4 Correlation matrix :	19
CHAPTER 2: Model Selection	20
1. Evaluation Metrics:	20
1.1. Score:	20
1.2. Roc Auc Score:	20

1.3. Confusion Matrix:	21
1.4. Sensitivity:	23
1.5. Specificity:	23
1.6. Precision:	23
2. Model Selection:	24
3. Logistic Regression:	24
CHAPTER 3: Model deployment	24
Deployment method:	24
User Interface:	25
Postman:	26
Webography	27

Table of Figures

<i>Fig 1: Benign Tumor Ultrasound image</i>	13
<i>Fig 2: Benign Tumor Eclipse shape</i>	13
<i>Fig 3: Malignant Tumor Ultrasound image</i>	13
<i>Fig 4: Malignant Tumor Eclipse shape</i>	13
<i>Fig 5: Percentage of Malignant/Benign in Dataset</i>	16
<i>Fig 6: Distribution of data via Histograms grouped by malignancy</i>	17
<i>Fig 7: Distribution of data via BoxPlots grouped by malignancy</i>	18
<i>Fig 8: Correlation matrix</i>	19
<i>Fig 9: model evaluation by Roc-Auc-Score and score</i>	20
<i>Fig 10: Confusion matrix</i>	21
<i>Fig 11: RandomForrest Confusion Matrix</i>	22
<i>Fig 12: GaussianNB Confusion Matrix</i>	22
<i>Fig 13: Decision Tree Confusion Matrix</i>	22
<i>Fig 14: kNN Classifier Confusion Matrix</i>	22
<i>Fig15: Logistic Regression Confusion Matrix</i>	22
<i>Fig 16: SVC Confusion Matrix</i>	22
<i>Fig 17 : metric evaluation for RMC , NB, DTC, kNN, LR and SVC</i>	23
<i>Fig 18: The web app UI</i>	25
<i>Fig 19: Postman test</i>	26

List of Abbreviations

EDA	Exploratory Data Analysis
SVM	Support Vector Machine
LR	Logistic Regression
M	Malignant
B	Benign
JS	JavaScript
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
API	Application Programming Interface
REST	REpresentational State Transfer
HTTP	HyperText Transfer Protocol
UI	User Interface
RFC	Random Forest Classifier
NB	Naive Bayesian
DTC	Decision Tree Classifier
kNN	k-Nearest Neighbors
SVC	Support Vector Classifier

Problem Statement

Breast cancer is a common cause of death for women worldwide. According to the global cancer statistics 2018, the incidence and mortality of cancer in China rank the first in the world, among which the incidence of breast cancer is the highest among women and the mortality rate ranks the fifth. Early detection, early diagnosis, and early treatment are the key to improve the recovery rate of breast cancer and reduce the mortality rate. Therefore, it is desired to develop an effective benign and malignant breast tumor classification method [1].[18]

So the data scientists will not just sit on the couch and watch, no !! They started looking for solutions. In this report, we will adopt one of these methods that is based on converting ultrasound images to extract mathematical and morphological measures so we can use them as features in our dataset.

CHAPTER 1: Exploratory Data Analysis (*EDA*)

Exploratory data analysis (*EDA*) is used by data scientists to analyze and investigate data sets and summarize their main characteristics, often employing data visualization methods.

1. Work Environment

1.1 Languages



Python

Is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation. Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library [2].



HTML

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets (CSS) and scripting languages such as JavaScript.

Web browsers receive HTML documents from a web server or from local storage and render the documents into multimedia web pages.

HTML describes the structure of a web page semantically and originally included cues for the appearance of the document.[3]



CSS

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML (including XML dialects such as SVG, MathML or XHTML).[1] CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.[4]



JavaScript

often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. As of 2022, 98% of websites use JavaScript on the client side for webpage behavior, often incorporating third-party libraries. All major web browsers have a dedicated JavaScript engine to execute the code on users' devices.[5]

1.2 Packages



Scikit-learn

Is a Python module for machine learning built on top of SciPy and is distributed under the 3-Clause BSD license.

The project was started in 2007 by David Cournapeau as a Google Summer of Code project, and since then many volunteers have contributed. See the About us page for a list of core contributors.

It is currently maintained by a team of volunteers [6].



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. It is free software released under the three-clause BSD license [7].



Matplotlib

Is a plotting library for the Python programming language and its numerical mathematics extension NumPy. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK [8].



Seaborn

Is a library for making statistical graphics in Python. It builds on top of **matplotlib** and integrates closely with **pandas** data structures.

Seaborn helps you explore and understand your data. Its plotting functions operate on dataframes and arrays containing whole datasets and internally perform the necessary semantic mapping and statistical aggregation to produce informative plots. Its dataset-oriented, declarative API lets you focus on what the different elements of your plots mean, rather than on the details of how to draw them [9].



NumPy

Is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays [10].



jQuery

Is a JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, CSS animation, and Ajax. It is free, open-source software using the permissive MIT License. As of Aug 2022, jQuery is used by 77% of the 10 million most popular websites [11].

1.3 Frameworks



Flask

Is a micro web framework written in Python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions [12].

2. Data description

2.1 About Dataset:

Breast cancer is the most common cancer amongst women in the world. It accounts for 25% of all cancer cases, and affected over 2.1 Million people in 2015 alone. It starts when cells in the breast begin to grow out of control. These cells usually form tumors that can be seen via X-ray or felt as lumps in the breast area.

The key challenge against its detection is how to classify tumors into malignant (cancerous) or benign(non cancerous). We ask you to complete the analysis of classifying these tumors using machine learning (with SVMs) and the Breast Cancer Wisconsin (Diagnostic) Dataset [13].

Acknowledgements:

This dataset has been referred to by Kaggle.

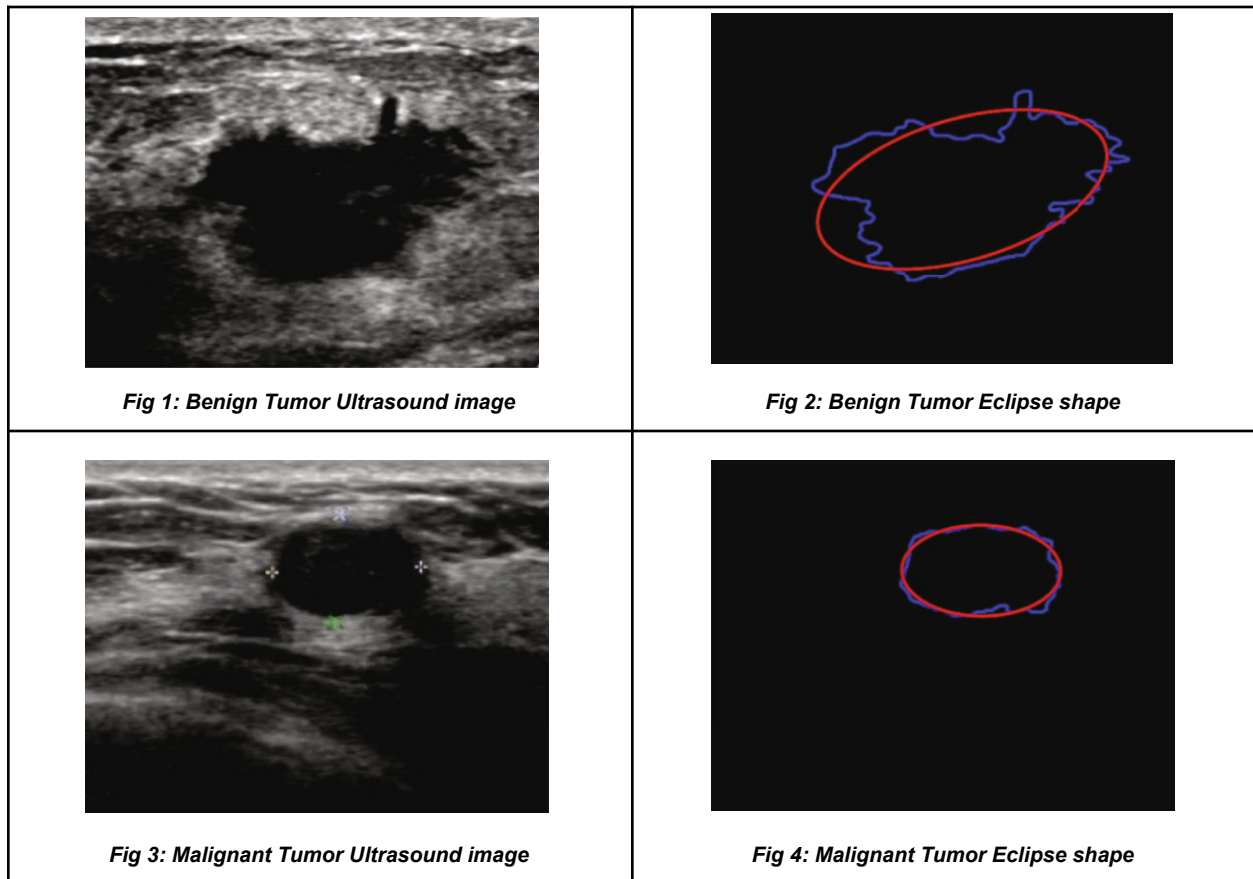
Objective:

- *Understand the Dataset & cleanup (if required).*

- ***Build classification models to predict whether the cancer type is Malignant or Benign.***
- ***Also fine-tune the hyperparameters & compare the evaluation metrics of various classification algorithms.***

2.2 Features:

From these ultrasound images, they have created the features of our dataset.



1. ID number
2. Diagnosis (M = malignant, B = benign)

Ten real-valued features are computed for each cell nucleus:

- a. radius (mean of distances from center to points on the perimeter)

The Radial distance can be calculated as follows:

$$D(t) = \sqrt{(p_t - x_0)^2 + (q_t - y_0)^2}$$

where the tumor edge points are denoted as $P_t(p_t, q_t)$ and the center point is denoted as (x_0, y_0) .

- b. texture (standard deviation of gray-scale values)
- c. perimeter
- d. area
- e. smoothness (local variation in radius lengths)
- f. compactness (perimeter² / area - 1.0)

Compactness measures the similarity between the shape of a breast tumor and its fitting circle. The closer the compactness value is to 1, the less likely the tumor is to be malignant, expressed as follows:

$$C = \frac{A}{4\pi L}$$

- g. concavity (severity of concave portions of the contour)
- h. concave points (number of concave portions of the contour)
- i. symmetry
- j. 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.

This analysis aims to observe which features are most helpful in predicting malignant or benign cancer and to see general trends that may aid us in model selection and hyper parameter selection in future analysis [14].

2.3 Descriptive statistics

2.3.1 Data dimension:

- The number of **features** is 32
- The number of **samples** is 569

2.3.2 Data Information:

#	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

dtypes: float64(30), int64(1), object(1)

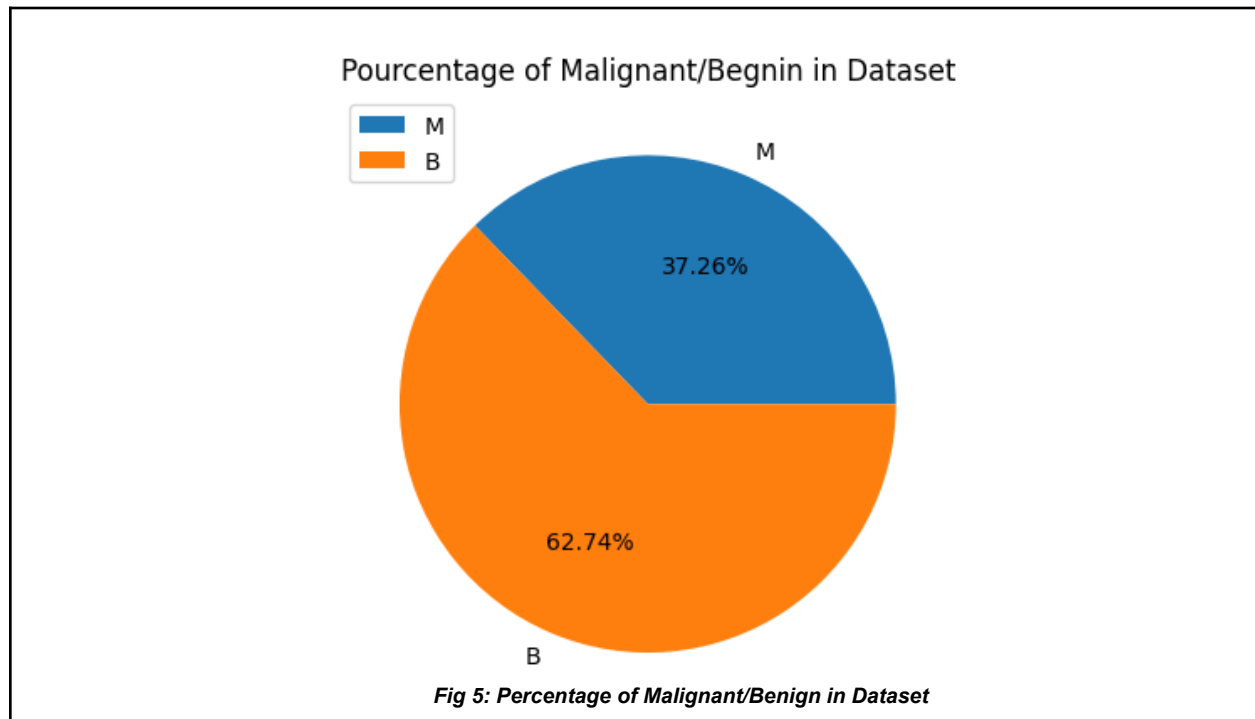
memory usage: 142.4+ KB

- As we can see, there are no NULL values in our data which is good news :)

2.4 Data Visualization:

One of the main goals of visualizing the data here is to observe which **features** are most helpful in **predicting malignant** or **benign** cancer. The other is to see **general trends** that may aid us in **model selection** and **hyperparameter selection**.

2.4.1 Distribution of the malignancy in the dataset:



Notice:

- We can see that benign tumors are the major category, which is normal.

2.4.2 Distribution of data via Histograms grouped by malignancy:

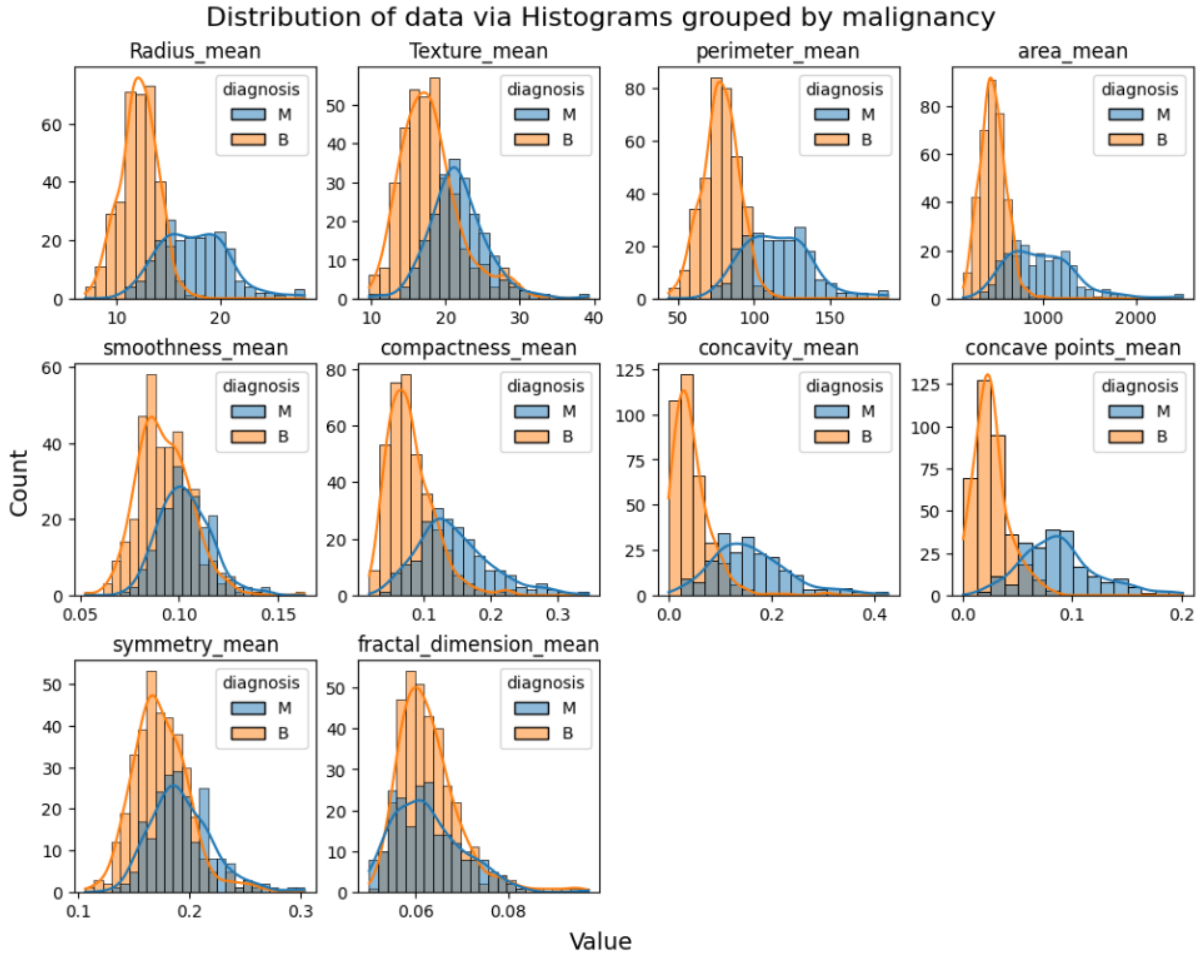


Fig 6 : Distribution of data via Histograms grouped by malignancy

Notice:

- We can see that we have a normal distribution: **Texture mean, Smoothness mean, symmetry mean**.
- All the features have a right skewness (**Positive Skewness**).
- Larger values have a **higher correlation** with **malignant** diagnosis.

2.4.3 Distribution of data via BoxPlots grouped by malignancy:

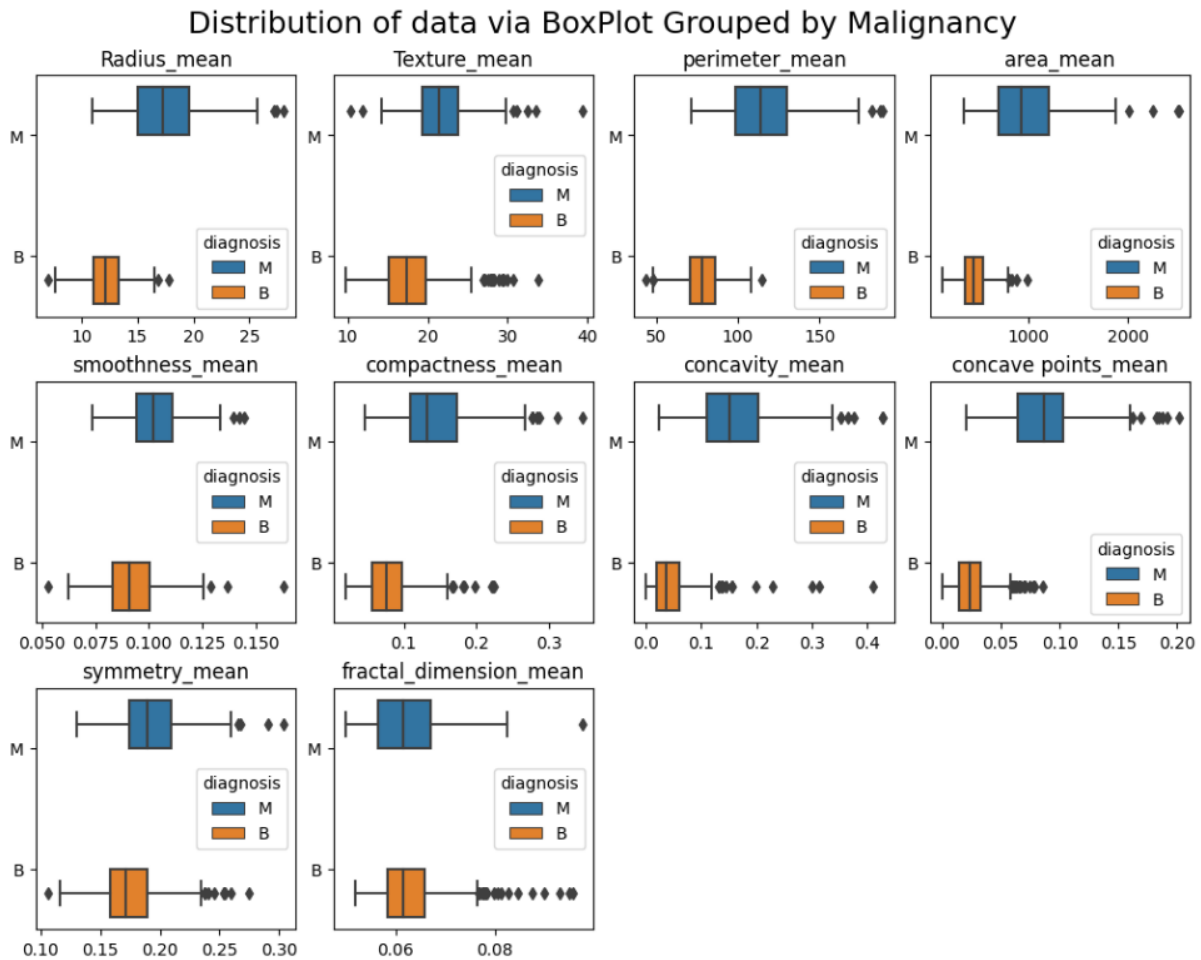
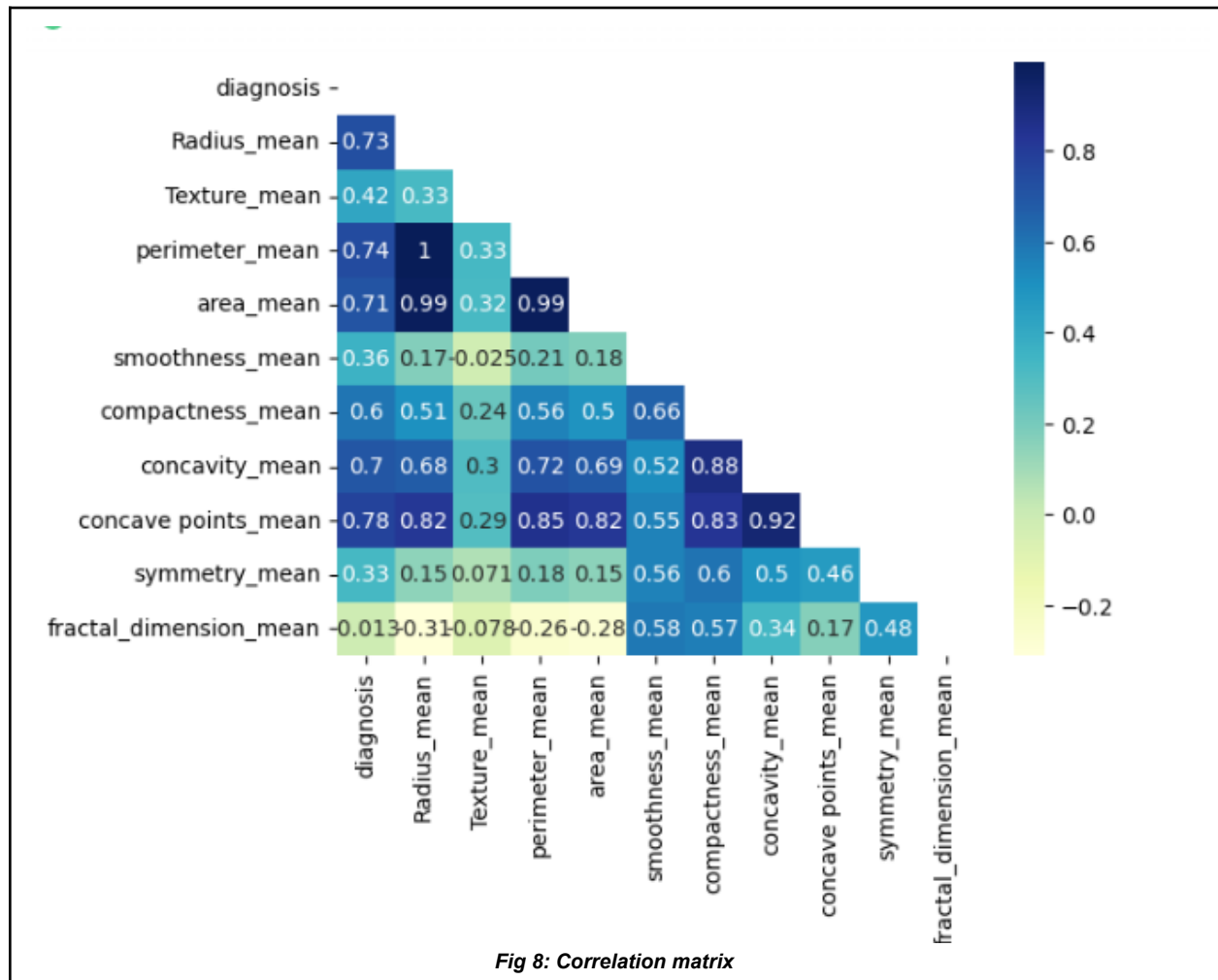


Fig 7: Distribution of data via BoxPlots grouped by malignancy

Notice:

- We can see that we have some outliers, but they do not impact the malignancy of a tumor.

2.4.4 Correlation matrix :



Observations

- Larger values of these parameters tend to show a correlation with malignant tumors.
- mean values of texture, smoothness, symmetry or fractal dimension does not show a particular preference of one diagnosis over the other.
- In all of the histograms there are no noticeable large outliers that need a further cleanup.

CHAPTER 2: Model Selection

Model selection is the process of selecting one final machine learning model from a collection of candidate machine learning models for a training dataset.

1. Evaluation Metrics:

1.1. Score:

Accuracy is then given as the number of correct predictions divided by the total number of predictions.[15]

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total Number of predictions}}$$

1.2. Roc Auc Score:

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the 'signal' from the 'noise'. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.[16]

models			
	Model object	Roc_Auc Score float64	Score float64
0	RandomForestClassifier	0.9795827123695975	0.9415204678362573
1	GaussianNB	0.9809239940387482	0.9532163742690059
2	DecisionTreeClassifier	0.8980625931445603	0.9064327485380117
3	KNeighborsClassifier	0.9728763040238451	0.9473684210526315
4	LogisticRegression	0.9906110283159464	0.9590643274853801
5	SVC	0.9839046199701937	0.9415204678362573

Fig 9 : model evaluation by Roc-Auc-Score and score

1.3. Confusion Matrix:

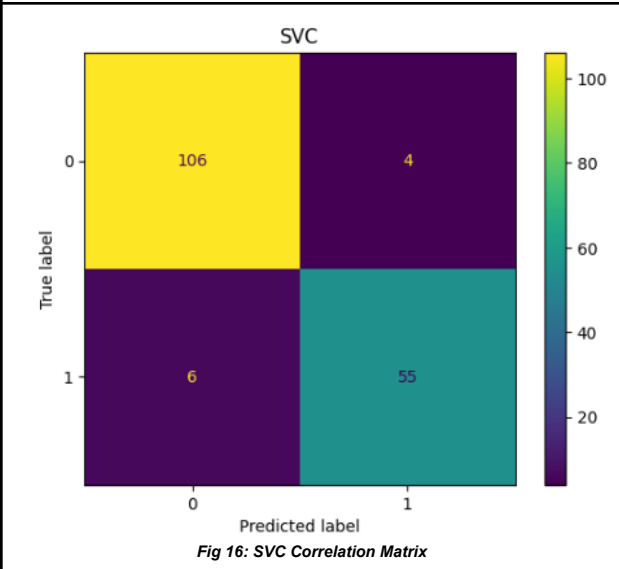
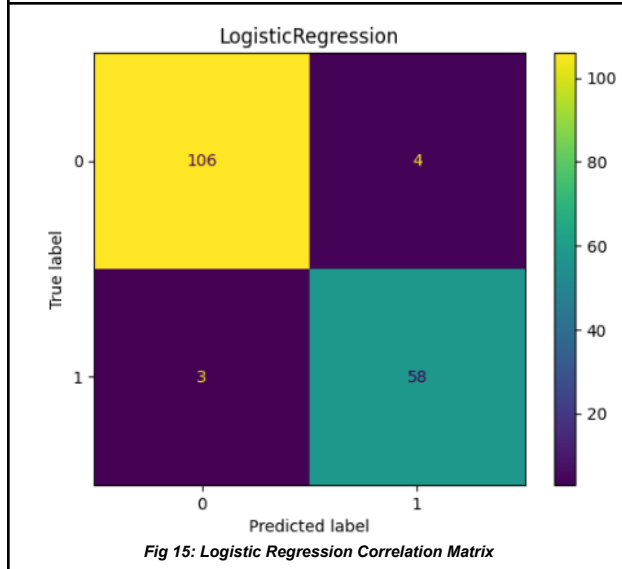
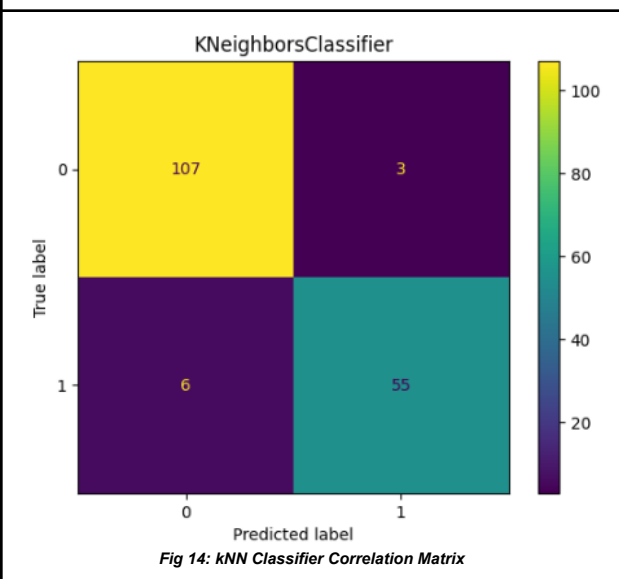
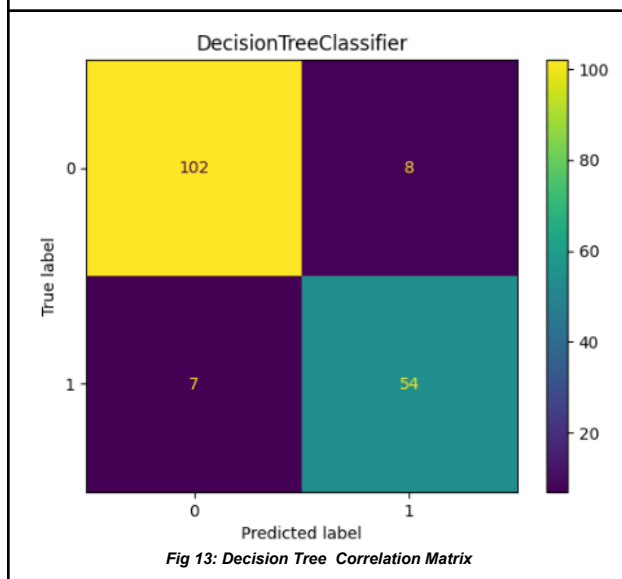
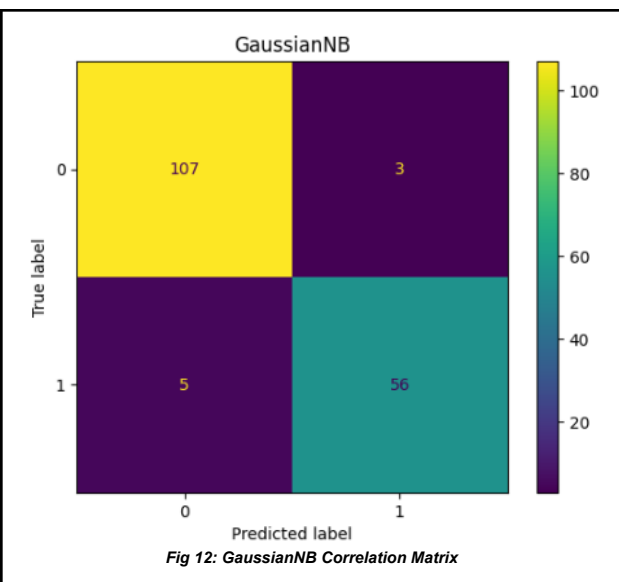
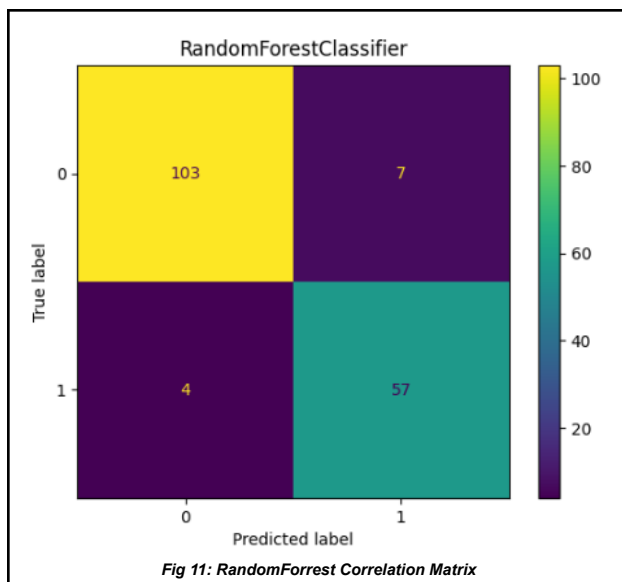
Confusion matrix is a performance measurement for machine learning classification problems where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.[17]

		Predicted condition	
Total population = P + N		Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP)	False negative (FN)
	Negative (N)	False positive (FP)	True negative (TN)

Fig 10 : Confusion matrix

Let's now define the most basic terms, which are whole numbers (not rates):

- **true positives (TP):** These are cases in which the model predicted Benign (they do not have the disease), and they do not have the disease.
- **true negatives (TN):** the model predicted Malignant, and the actual condition is Malignant
- **false positives (FP):** the model predicted Benign, but the actual condition is Malignant. (Also known as a "Type I error.")
- **false negatives (FN):** the model predicted Malignant, but the actual condition is Benign. (Also known as a "Type II error.")



1.4. Sensitivity:

Sensitivity (true positive rate) refers to the probability of a positive test, conditioned on truly being positive. Sensitivity refers to the test's ability to correctly detect ill patients who do have the condition. In the example of a medical test used to identify a condition, the sensitivity (sometimes also named the detection rate in a clinical setting) of the test is the proportion of people who test positive for the disease among those who have the disease. Mathematically, this can be expressed as [18]:

$$Sensitivity = \frac{TP}{TP + FN}$$

1.5. Specificity:

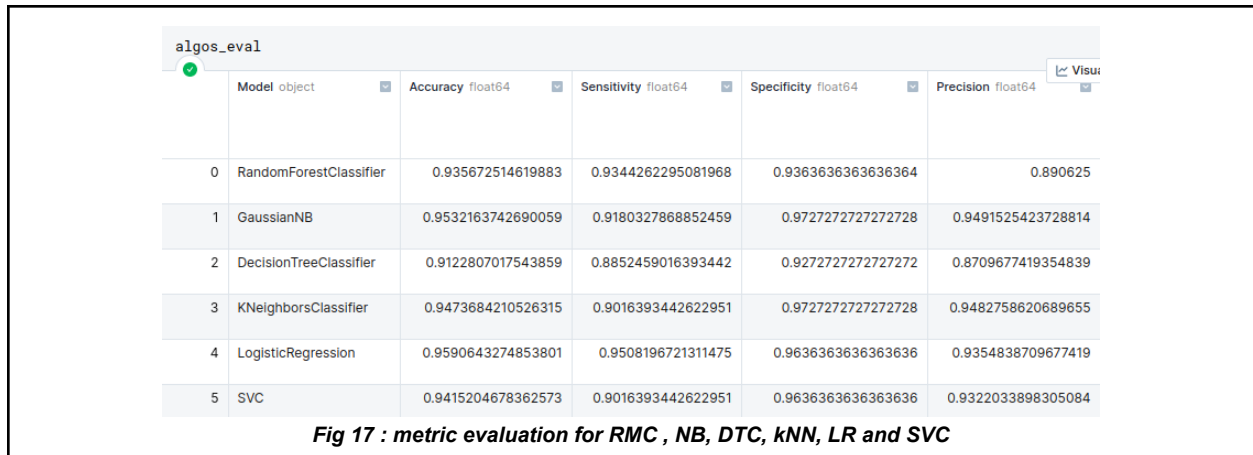
Specificity (true negative rate) refers to the probability of a negative test, conditioned on truly being negative. Specificity relates to the test's ability to correctly reject healthy patients without a condition. Specificity of a test is the proportion of those who truly do not have the condition who test negative for the condition. Mathematically, this can also be written as:

$$Specificity = \frac{TN}{TN + FP}$$

1.6. Precision:

Precision is a measure of how many of the positive predictions made are correct (true positives). The formula for it is:

$$Precision = \frac{TP}{TP + FP}$$



	Model object	Accuracy float64	Sensitivity float64	Specificity float64	Precision float64
0	RandomForestClassifier	0.935672514619883	0.9344262295081968	0.9363636363636364	0.890625
1	GaussianNB	0.9532163742690059	0.9180327868852459	0.9727272727272728	0.9491525423728814
2	DecisionTreeClassifier	0.9122807017543859	0.8852459016393442	0.9272727272727272	0.8709677419354839
3	KNeighborsClassifier	0.9473684210526315	0.9016393442622951	0.9727272727272728	0.9482758620689655
4	LogisticRegression	0.9590643274853801	0.9508196721311475	0.9636363636363636	0.9354838709677419
5	SVC	0.9415204678362573	0.9016393442622951	0.9636363636363636	0.9322033898305084

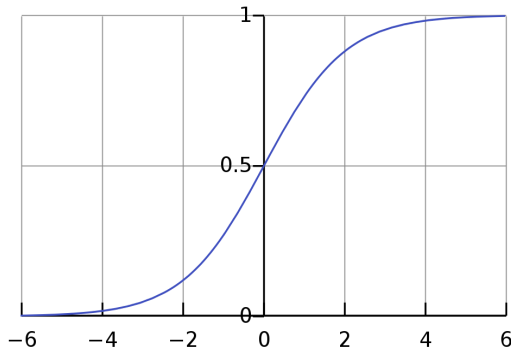
Fig 17 : metric evaluation for RMC , NB, DTC, kNN, LR and SVC

2. Model Selection:

As a start we splitted out data into a train and test dataset, then we fitted six of the most popular ML classification algorithms - **RFC**, **NB**, **DTC**, **kNN**, **LR** and **SVC** (see abbreviation table) to the training dataset, and measured this algorithms performance on test dataset using the evaluation metrics.

As shown in the figures 11, 12, 13, 14, 15, 16 and 17, the **Logistic Regression** is the most performant algorithm, *the most dangerous mistake* that our model should not make is to predict a *Malignant* tumor as a *Benign* one. Therefore, we use it in the deployment phase of our project.

3. Logistic Regression:



Logistic regression is a statistical analysis method to predict a binary outcome, such as yes or no, based on prior observations of a data set.

A logistic regression model predicts a dependent data variable by analyzing the relationship between one or more existing independent variables. For example, a logistic regression could be used to predict whether a political candidate will win or lose an election or whether a high school student will be admitted or not to a particular college. These binary outcomes allow straightforward decisions between two alternatives.[19]

CHAPTER 3: Model deployment

Deployment method:

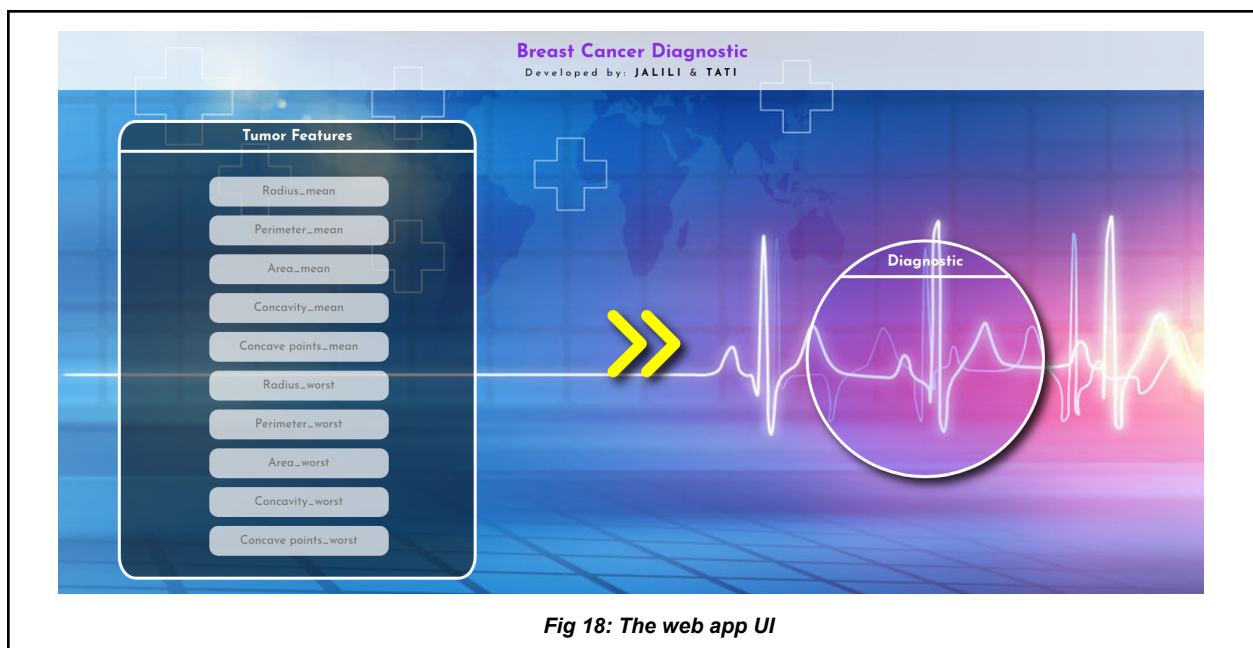
The idea is to deploy the model over a web application using **Flask server** as an API, in addition to basic **HTML**, **CSS** and **JS** codes to structure & design our web app. The model has been deployed as a pickle file, then used to predict the malignancy of the tumor inserted in the app.

To load a saved model from a Pickle file, all you need to do is pass the “pickled” model into the Pickle load() function and it will be deserialized. By assigning this back to a model object, you can then run your original model's predict() function, pass in some test data and get back an array of predictions.

Also, we used **Postman** which is a software that helps us to test the HTTP requests between the Flask server & the features in the front-end. Since it is a micro-framework, it is very easy to use and lacks most of the advanced functionality which is found in a full-fledged framework. Therefore, building a **REST API in Flask is very simple**.

This website is destined to doctors to help them diagnose a patient by inserting the tumor data.

User Interface:



So, on the left of the screen you can insert tumor data, then after clicking on the yellow arrows you will get your diagnostic on the right of the screen, just that simple.

Postman:

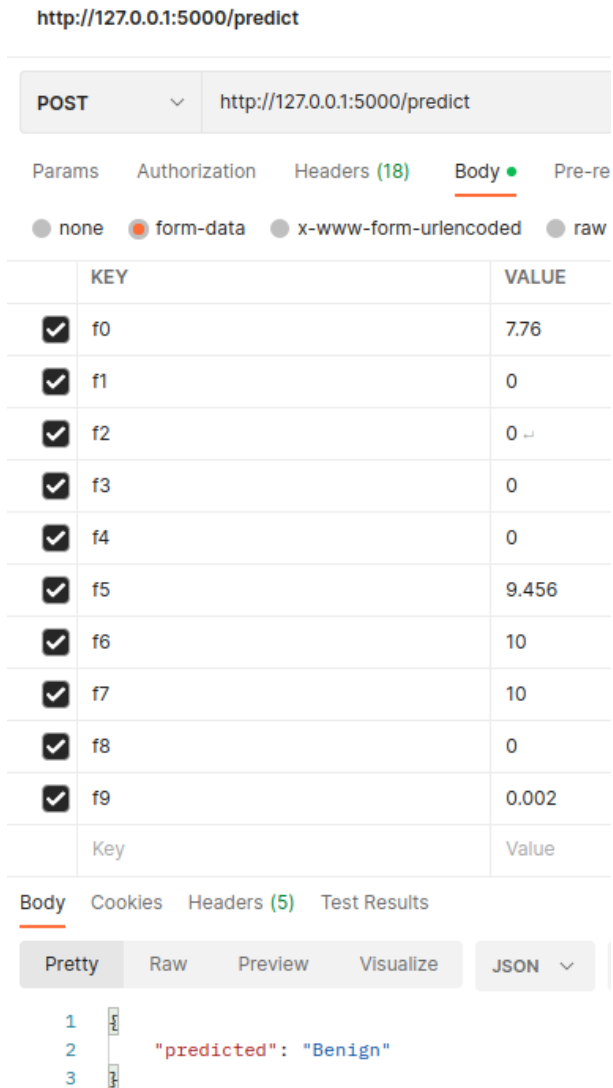


Fig 19: Postman test

Well, there are a lot of request's types. One of them is **POST**. In web services, POST requests are **used to send data to the API server to create or update a resource**. The data sent to the server is stored in the request body of the HTTP request. The simplest example is a contact form on a website.

Keys:

- F0: Radius mean*
- F1: Perimeter mean*
- F2: Area mean*
- F3: Concavity mean*
- F4: Concave Points mean*
- F5: Radius worst*
- F6: Perimeter worst*
- F7: Area worst*
- F8: Concavity worst*
- F9: Concave Points worst*

Response:

`sonify()` function returns a Response object. Flask serializes your data as JSON and adds it to this Response object. It also adds the appropriate mime type by setting the `content-type` header field to `application/json` [20].

Webography

- [1] [A Benign and Malignant Breast Tumor Classification Method via Efficiently Combining Texture and Morphological Features on Ultrasound Images](#)
- [2] [Python \(programming language\) - Wikipedia](#)
- [3] [HTML - Wikipedia](#)
- [4] [CSS - Wikipedia](#)
- [5] [JavaScript - Wikipedia](#)
- [6] [scikit-learn: machine learning in Python](#)
- [7] [pandas \(software\) - Wikipedia](#)
- [8] [Matplotlib - Wikipedia](#)
- [9] [An introduction to seaborn](#)
- [10] [NumPy - Wikipedia](#)
- [11] [jQuery - Wikipedia](#)
- [12] [Flask \(web framework\) - Wikipedia](#)
- [13] [Breast Cancer Dataset | Kaggle](#)
- [14] [Breast cancer data analysis](#)
- [15] [Receiver operating characteristic - Wikipedia](#)
- [16] [AUC-ROC Curve in Machine Learning Clearly Explained - Analytics Vidhya](#)
- [17] [Understanding Confusion Matrix | by Sarang Narkhede | Towards Data Science](#)
- [18] [Sensitivity and specificity - Wikipedia](#)
- [19] [What is Logistic Regression? - Definition from SearchBusinessAnalytics](#)
- [20] [Difference Between `json.dumps\(\)` and `flask.jsonify\(\)` | Sentry](#)

END.