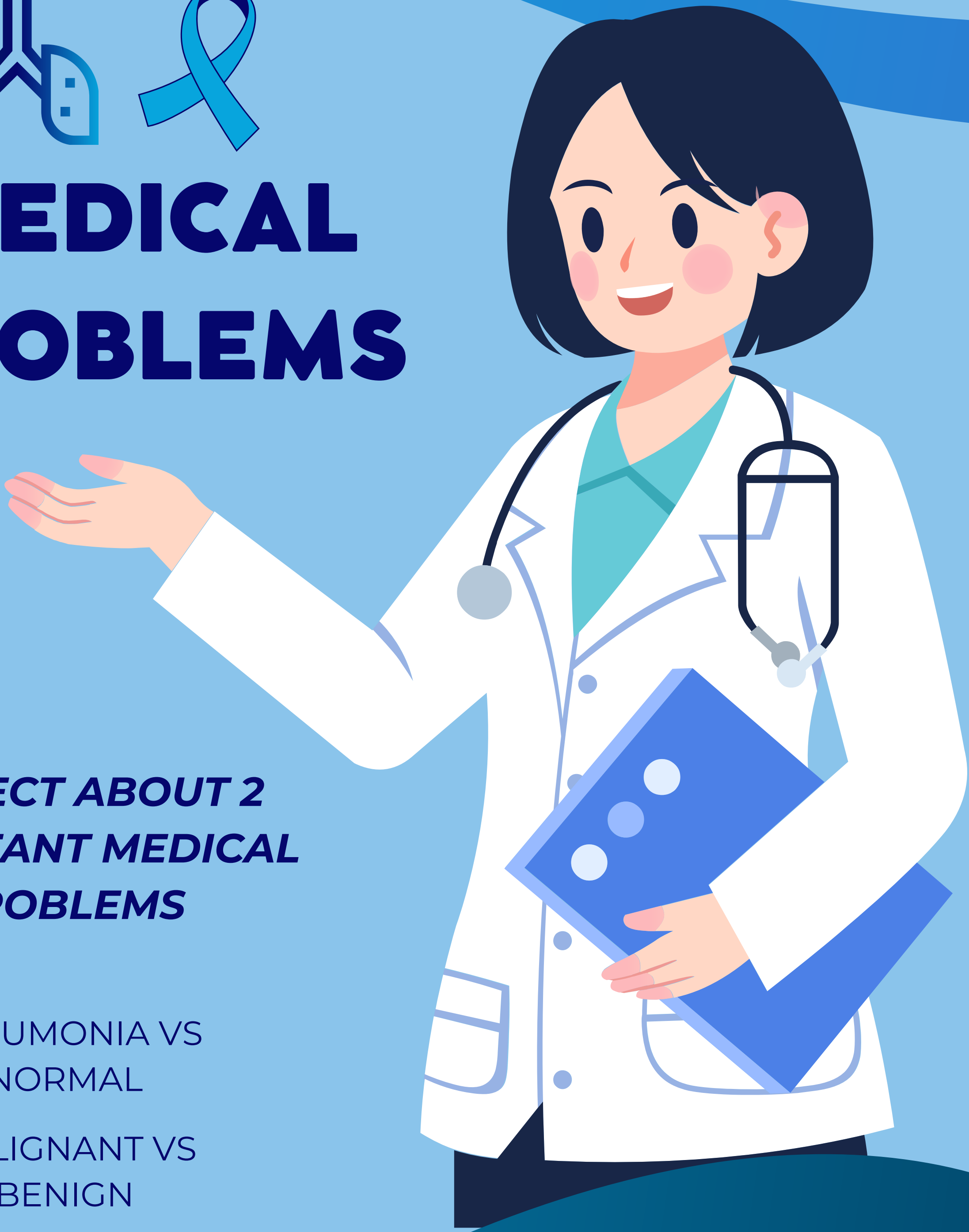
MEDICAL PROBLEMS



PROJECT ABOUT 2 IMOPRTANT MEDICAL PROBLEMS

- PNEUMONIA VS NORMAL
- MALIGNANT VS BENIGN

Mahmoud Essam
Abdelrhman Ashraf
Abdullah Hussien
Fares Mohammed
Zyad Ashraf

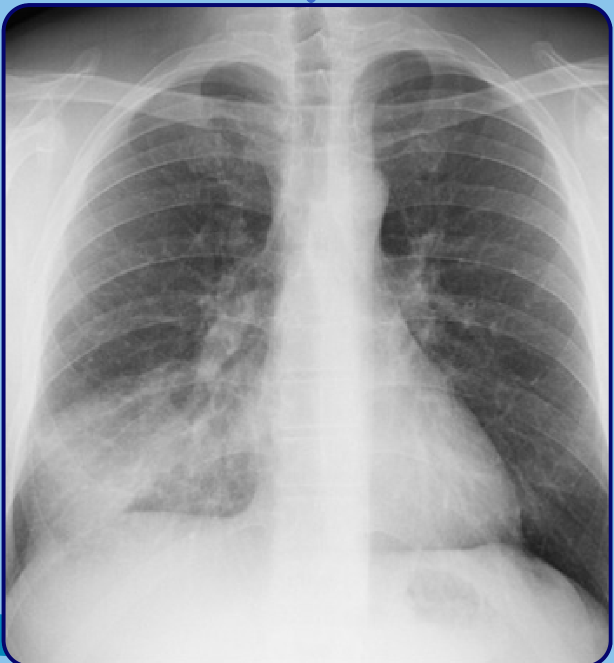
In our project , We have 2 datasets

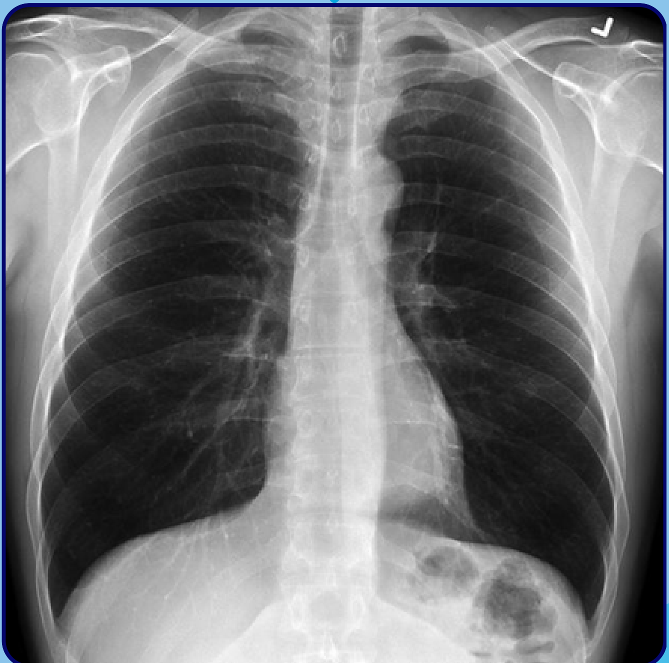
1-Chest X-Ray Images (Pneumonia VS Normal)
2-Breast Cancer Data (Malignant vs Benign)

Chest X-Ray Images

We Used a dataset from Kaggle.com.
Source: ***Chest X-Ray Images (Pneumonia).***

We have a total of 5,863 X-ray images divided into 2 categories [Pneumonia,Normal].





Our Goal is to build models that will be able to classify the X-ray image either it's **Normal** **or** **Pneumonia.**

We did this by building Several models

SVM

SVM (Support Vector Machine) is a supervised machine learning model used for classification and regression tasks.
It works by finding the best possible line or hyperplane that separates different classes in a dataset.

Logistic regression

Logistic regression is a statistical method used for binary classification tasks , Where the outcome variable is categorical and has two possible classes.
Despite its name, Logistic regression is primarily employed for classification rather than regression.

CNN

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for processing structured grid data, such as images.
CNNs have proven to be highly effective in image-related tasks, including image classification, object detection, and image segmentation.

VGG16

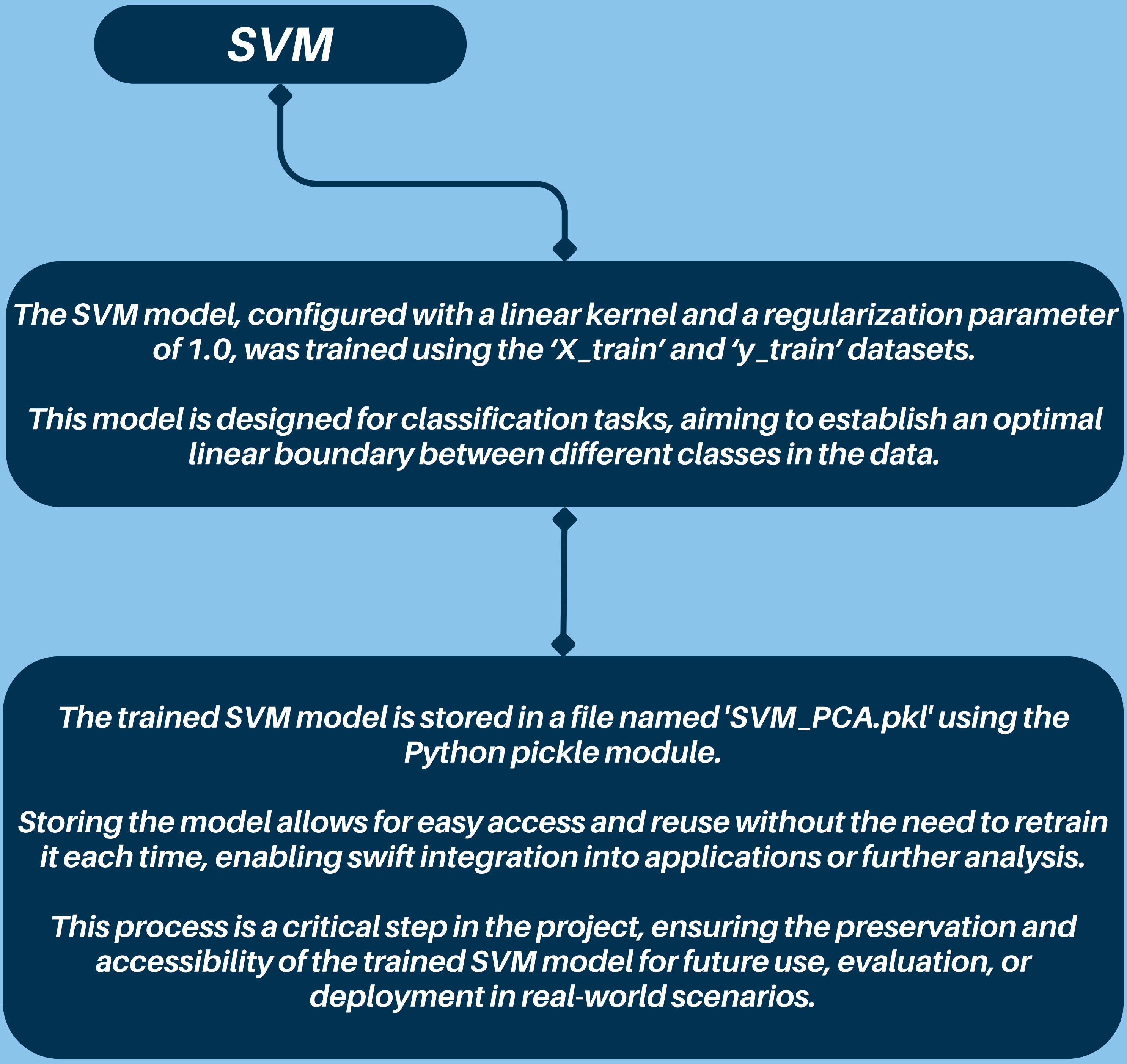
VGG16 is a convolutional neural network (CNN) architecture that gained prominence for its simplicity and remarkable performance in image classification.
The (16) refers to the total number of weight layers used in the model.

Mobilenet

Developed by Google, is a lightweight CNN designed for mobile and embedded vision.
It prioritizes efficiency and accuracy in image classification through depthwise separable convolutions and a streamlined architecture.

Model	Test Accuracy
SVM	74.19%
Logistic Regression	78.52%
CNN	90.22%
VGG16	91.83%
MobileNet	91.99%

Models Details



SVM

The SVM model, configured with a linear kernel and a regularization parameter of 1.0, was trained using the 'X_train' and 'y_train' datasets.

This model is designed for classification tasks, aiming to establish an optimal linear boundary between different classes in the data.

The trained SVM model is stored in a file named 'SVM_PCA.pkl' using the Python pickle module.

Storing the model allows for easy access and reuse without the need to retrain it each time, enabling swift integration into applications or further analysis.

This process is a critical step in the project, ensuring the preservation and accessibility of the trained SVM model for future use, evaluation, or deployment in real-world scenarios.

Type	Normal	Pneumonia
Normal	82	152
Pneumonia	9	381

Interpretation:

- ***True Positive** (TP): 381 indviduals were correctly classified as having pneumonia*
- ***True Negatives** (TN): 82 individuals were correctly classified as not having pneumonia.*
- ***False Positives** (FP): 152 individuals were incorrectly classified as having pneumonia when they did not.*
- ***False Negatives** (FN): 9 individuals were incorrectly classified as not having pneumonia when they did.*

Additional Considerations:

- ***Purpose**: The specific purpose of this matrix in our project will guide how we analyze and interpret the data.*
- ***Context**: Understanding the context in which the data was collected is crucial for accurate interpretation.*
- ***Normalization**: If the raw counts represent different proportions of the total population, normalization may be necessary for comparisons.*
- ***Visual Representation** : The matrix effectively visualizes the distribution of individuals across the two categories.*

Models Details

Logistic Regression

Logistic regression is a good fit for this dataset because it's easy to understand how it makes predictions. It works well when we have a moderate amount of data, which is common in medical situations. This simplicity and its ability to explain why it makes certain predictions make it a sensible choice for starting the analysis of pneumonia detection.

1. **Initial Setup:** First, a logistic regression model is set up by the line log_reg = LogisticRegression(). This model is like a tool that helps us predict or classify things based on given information.
2. **Training:** The fit() function is used to teach this model. It learns from a set of examples where we already know both the input (X_train - features like patient details) and the output (y_train - what we want to predict, like whether a patient has pneumonia or not).
3. **Making Predictions:** Once the model has learned from the training data, it's put to the test. We use X_test (which contains features of new, unseen examples) to ask the model to predict the outcomes. The predict() function helps us get these predictions.

Once this model is trained and has learned from this information, it's saved into a file named 'LR_PCA.pkl'. This way, we can use this trained model later on without needing to train it again from scratch every time we need to make predictions.

Type	Normal	Pneumonia
Normal	116	118
Pneumonia	16	374

Interpretation:

- *True Positive (TP): 374 indviduals were correctly classified as having pneumonia*
- *True Negatives (TN): 116 individuals were correctly classified as not having pneumonia.*
- *False Positives (FP): 118 individuals were incorrectly classified as having pneumonia when they did not.*
- *False Negatives (FN): 16 individuals were incorrectly classified as not having pneumonia when they did.*

Additional Considerations:

- *Purpose: The specific purpose of this matrix in our project will guide how we analyze and interpret the data.*
- *Context: Understanding the context in which the data was collected is crucial for accurate interpretation.*
- *Normalization: If the raw counts represent different proportions of the total population, normalization may be necessary for comparisons.*
- *Visual Representation : The matrix effectively visualizes the distribution of individuals across the two categories.*

Models Details

CNN

We defined (CNN) model using the TensorFlow and Keras framework for image classification.

1. *Input Layer:*
   - *Conv2D layer with 32 filters, a filter size of (3,3), ReLU activation function, and input shape of (200, 200, 3), indicating images of size 200x200 pixels with 3 color channels (RGB).*
2. *Pooling Layer:*
   - *MaxPooling2D layer with a pool size of (2,2) to downsample the spatial dimensions.*
3. *Second Layer:*
   - *Conv2D layer with 64 filters and a filter size of (3,3), followed by another MaxPooling2D layer.*
4. *Dropout Layer:*
   - *Dropout layer with a dropout rate of 0.2, which helps prevent overfitting by randomly setting a fraction of input units to zero during training.*
5. *Third Layer:*
   - *Conv2D layer with 128 filters and a filter size of (3,3), followed by MaxPooling2D and Dropout layers.*
6. *Fourth Layer:*
   - *Conv2D layer with 256 filters and a filter size of (3,3), followed by MaxPooling2D layer.*
7. *Flattening Layer:*
   - *Flatten layer to convert the 3D output to a 1D vector to prepare for the fully connected layers.*
8. *Dropout Layer:*
   - *Another Dropout layer with a dropout rate of 0.2.*
9. *Dense Layers:*
   - *Dense (fully connected) layer with 256 units and ReLU activation function.*
   - *Output layer with 1 unit and a sigmoid activation function, suitable for binary classification tasks.*

And then we set up data generators for training, validation, and testing images.
The training generator includes data augmentation to enhance the diversity of the training set.
The validation generator is used for assessing the model's performance on unseen data, and the test generator is for evaluating the final model on a separate test set.

Models Details

VGG16

We used VGG16 as a pre-trained model and removed the used top layers to add top layers that suite our problem.
We imported the 16 hidden layers that were already trained from the original model , and then we added the top layers.

1. *Flatten Layer:*
   - *This layer transforms the output from the previous layers into a one-dimensional vector. It "flattens" the multi-dimensional output into a format suitable for feeding into a fully connected neural network.*
2. *Dense Layer (512 units, ReLU activation):*
   - *The flattened output is then passed through a dense layer with 512 units. This layer introduces non-linearity to the model using the Rectified Linear Unit (ReLU) activation function. It helps capture complex relationships in the data.*
3. *Dropout Layer (Dropout Rate: 0.5):*
   - *Dropout is a regularization technique that randomly drops a fraction of the input units to prevent overfitting during training. In this case, 50% of the units are randomly set to zero during each update, enhancing the model's generalization capability.*
4. *Dense Layer (1 unit, Sigmoid activation):*
   - *The final dense layer consists of one unit with a sigmoid activation function. This setup is common for binary classification tasks. The sigmoid activation outputs a value between 0 and 1, representing the probability of the input belonging to the positive class (in this case, the presence of a specific feature).*

Then we made a block of code that generates a random photo from the data each time we run it and prints the prediction of it showing the confidence level of prediction.

Models Details

MobileNet

As we did in the VGG16 , We used MobileNet as a pre-trained model by removing the top layers and using the trained hidden layers.

And then we customized the top layers as we did with VGG16.

On top of MobileNet, a custom classifier is added, and the entire model is compiled for training.

Data augmentation is applied to the training set to prevent overfitting.

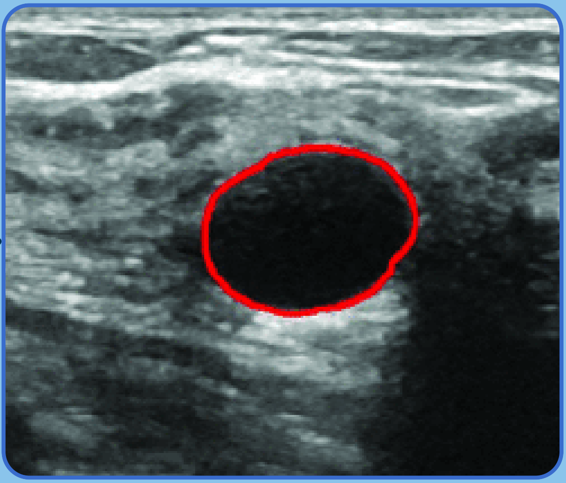
1. *Global Average Pooling Layer:*
   - *This layer reduces the spatial dimensions of the input and computes the average value for each feature map.*
   - *It helps in simplifying the model, reducing the number of parameters, and capturing essential information from the MobileNet's output.*
2. *Dense Layer (512 units, ReLU activation):*
   - *The global average-pooled output is passed through a dense layer with 512 units and a Rectified Linear Unit (ReLU) activation function.*
   - *This introduces non-linearity and helps the model capture more complex patterns in the data.*
3. *Dropout Layer (Dropout Rate: 0.5):*
   - *Dropout is applied to prevent overfitting during training.*
   - *It randomly sets 50% of the input units to zero during each update, enhancing the model's ability to generalize well to new, unseen data.*
4. *Output Layer (1 unit, Sigmoid activation):*
   - *The final dense layer consists of one unit with a sigmoid activation function.*
   - *This is suitable for binary classification tasks.*
   - *The sigmoid activation outputs a value between 0 and 1, representing the probability of the input belonging to the positive class (in this case, the presence of a specific feature).*

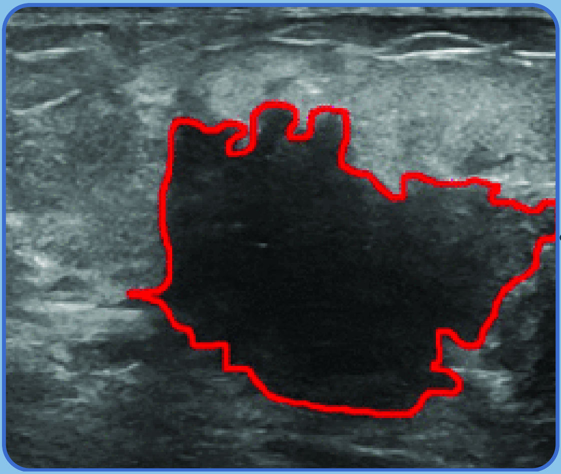
And we made the same block of code that generates a random photo printing it’s prediction with the level of confidence of this prediction.

Breast Cancer

Discovering a lump in the breast can be either harmless (benign), like a non-threatening bump, or serious (malignant), indicating cancer.
If it's okay, it's not cancer and poses no danger.
If it's not okay, and there's a concern it might be cancer, it's crucial to see the doctor promptly.
Quick medical attention is important to understand what's happening and receive the right assistance.

Benign





malignant

The features come from a picture of a breast lump taken with a computer. They tell us about the parts inside the cells in the pictures.

Attribute	Definition
ID	Identification number assigned to each data point
Diagnosis	Classification of the tumour as either malignant (M) or benign (B)
Mean Radius	Mean distance from the centre to points on the perimeter
Texture	Standard deviation of grey-scale values
Perimeter	Perimeter of the tumour
Area	Area covered by the tumour
Smoothness	Local variation in radius lengths
Compactness	Measure of how compact the shape of the tumour is
Concavity	Severity of concave portions of the contour
Concave Points	Number of concave portions of the contour
Symmetry	Symmetry of the tumour
Fractal Dimension	"Coastline approximation" - 1

What Suppose to happen among this data ?

I’ve built about 4 techniques, testing by it, if there will happen change in accuracy for data or not, so we have tried 4 techniques.

1. Principal Component Analysis
2.Kruskal Wallis Test
3.Chi Square Test
4.Without Any Technique

Principal Component Analysis (PCA) is important as it simplifies complex data by identifying and highlighting the most relevant patterns, making it easier to understand and analyze.

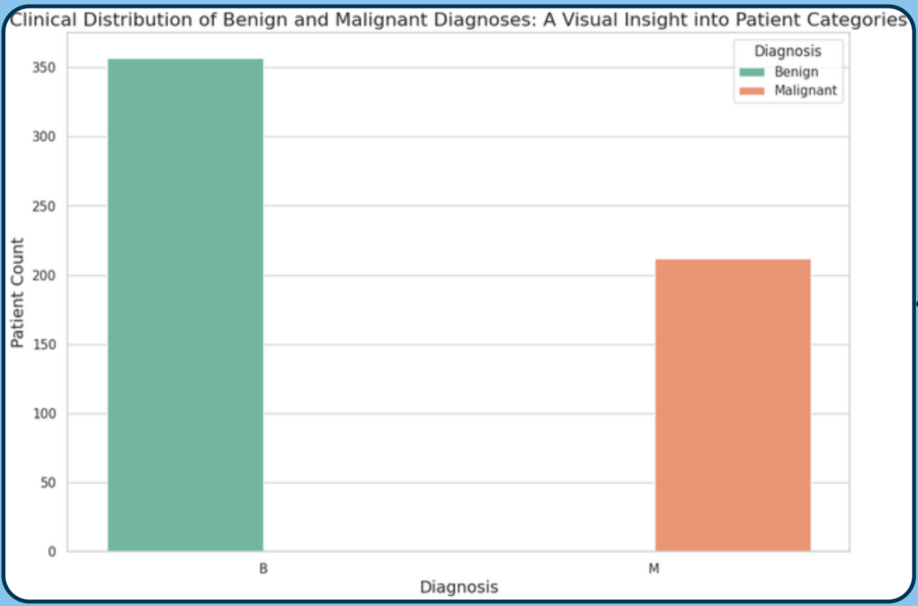
We have assigned about 99.9% of threshold for variance in data

What Suppose to happen among this data ?

The Kruskal-Wallis test is a non-parametric statistical test used to determine if there are any significant differences between the medians of three or more independent groups as ordinal data, providing an alternative to the one-way analysis of variance (ANOVA) for non-normally distributed data.

The Chi-Square test is a statistical method used to determine if there is a significant association between categorical variables on nominal data by comparing observed and expected frequencies.

📊 Clinical Insight: Patient Percentages 📊
✨ Percent of Benign Diagnoses: 63%
✨ Percent of Malignant Diagnoses: 37%



Used Models, and Checking Process before and after performing the techniques

Applying PCA Technique

after performing the decomposition, I have found that the data from 31 column, to 3 PCA Columns

Model	Accuracy on Train	Accuracy on Test
Logistic	94.29%	92.11%
SVM	94.51%	88.60%
Tree	100.00%	89.47%
Random Forest	100.00%	91.23%
NN	-	-

there's only 6 people classified as benign, but they have the disease as optimal Scenario

Used Models, and Checking Process before and after performing the techniques

Kruskal Wallis Test

as it as nominal data, but I wanted to check if there exist great difference between models in accuracy based on type of target or not

Model	Accuracy on Train	Accuracy on Test
Logistic	98.68%	95.61%
SVM	98.24%	95.61%
Tree	99.12%	92.11%
Random Forest	100.00%	93.86%
NN	-	-

model has improved to be 5 for now,

Used Models, and Checking Process before and after performing the techniques

Chi Square Test

as it as nominal data again, now will see if there exist great difference now or not based on scores

Model	Accuracy on Train	Accuracy on Test
Logistic	97.14%	95.61%
SVM	98.68%	95.61%
Tree	100.00%	92.11%
Random Forest	100.00%	93.86%
NN	-	-

and also this time best model achieved only 5 wrong classification

Used Models, and Checking Process before and after performing the techniques

Without Anything?

yep, without any modifications! just simple normal data

Model	Accuracy on Train	Accuracy on Test
Logistic	98.90%	94.74%
SVM	98.46%	95.61%
Tree	99.12%	93.86%
Random Forest	100.00%	92.98%
NN	98.90%	96.49%

Also Classified as 5, so in general there are 3% of people getting classified wrong as normal, but they have the disease

Name	ID
Mahmoud Essam Fathy	20221460231
Abdullah Hussien Ibraheem	20221427861
Abdelrahman Ashraf Ragab	20221374041
Fares Mohamed Fathy	20221461330
Zyad Ashraf Hafez	20221374025