

Step by step explanantion about project

Imagine you are explaining this project to someone who does not know anything about technology:

---

## What Have We Done? (In Simple Language)

We have built a smart tool called **BREAST DETECT-X** that helps doctors find out early if a person might have breast cancer. It does this by looking at certain measurements from tests (like the size and shape of a lump in the breast) and tells if the lump is likely to be safe (benign) or dangerous (malignant). But, unlike most computer tools, ours also explains why it made that decision, so doctors can trust and understand the results<sup>4123</sup>.

---

## Why Is This Needed?

- **Early Detection Saves Lives:** If breast cancer is found early, treatment is much easier and people have a much better chance of surviving<sup>9121</sup>.
  - **Current Problems:** Traditional tests (like mammograms or ultrasounds) can be slow, expensive, and sometimes hard to understand—especially for certain people or in places with fewer resources<sup>14</sup>.
  - **Trust in Computers:** Many computer tools can give answers, but they don't explain how they got those answers. Doctors need to know “why” before they trust a machine with someone's health<sup>142</sup>.
- 

## What Did We Actually Do?

### 1. Collected Data

- We used a real medical dataset that has information about breast lumps from hundreds of patients. Each record includes measurements like how wide, rough, or bumpy the lump is, and whether it turned out to be cancer or not<sup>15</sup>.

### 2. Cleaned and Prepared the Data

- We removed unnecessary information and made sure all the data was correct and in the right format for the computer to understand<sup>14</sup>.
- We adjusted the numbers so that no single measurement would unfairly affect the results (this is called normalization)<sup>1</sup>.

### 3. Trained a Computer Model

- We taught a computer program (called SVM, or Support Vector Machine) to look at the measurements and learn how to tell the difference between safe and dangerous lumps<sup>10</sup>.
- We made sure it could handle tricky cases where the differences are not obvious.

### 4. Made It Explainable

- We added special tools (called SHAP and LIME) that help the computer show which measurements were most important in making its decision. This way, the doctor doesn't just get an answer—they also get a reason why<sup>12</sup>.

### 5. Built a Simple Dashboard

- We created an easy-to-use webpage where a doctor (or even a patient) can enter the measurements, click a button, and get the prediction along with a simple explanation and graphs<sup>74</sup>.

---

## How Does It Work? (Step-by-Step)

1. A doctor enters the measurements of a lump into our tool.
2. The computer checks these numbers against what it has learned from past cases.
3. It predicts if the lump is likely to be safe or dangerous.
4. It shows the doctor which measurements mattered most for this prediction.
5. The doctor can see graphs and explanations to help them understand and trust the result.

---

## In Short

- We made a tool that helps find breast cancer early, explains its answers, and is easy for doctors to use.
- It helps save lives by making diagnosis faster, clearer, and more trustworthy.

---

## Why Is This Better?

- Doctors get both the answer and the explanation, so they can make better decisions.
- The tool is fast, easy to use, and fair—it works for everyone, regardless of background.

- It can be improved in the future by adding more data, like genetic or environmental factors<sup>142</sup>.
- 

### Summary:

We built a “smart assistant” for doctors to help spot breast cancer early and explain its thinking, making medical decisions safer and easier for everyone.

---

## Detailed Explanation with Definitions

### What is Breast Cancer?

**Breast cancer** is a disease where cells in the breast grow uncontrollably. There are two main types of breast tumors:

- **Benign Tumor:** A non-cancerous growth that does not invade nearby tissues or spread to other parts of the body.
- **Malignant Tumor:** A cancerous growth that can invade surrounding tissues and spread to other parts of the body (this spreading is called metastasis)

**Early detection** of breast cancer is crucial because it increases the chances of successful treatment and survival

---

### Why is Early Detection Needed?

- **Improved Survival:** Finding cancer early means treatment can start sooner, often before the cancer has spread.
  - **Better Outcomes:** Early-stage cancers are easier to treat and have higher survival rates.
  - **Awareness and Education:** Many people do not recognize the early signs of breast cancer. Tools that help with early detection can save lives
- 

### What Did We Build?

We developed an **interactive web application** using a method called **Support Vector Machine (SVM)** to help predict whether a breast tumor is benign or malignant based on measurements from medical tests (like how big or rough the lump is)

---

# Step-by-Step Explanation

## 1. Collecting and Understanding the Data

- **Dataset:** We used the Wisconsin Breast Cancer dataset, which contains measurements from breast tumor samples. Each sample has several features (like radius, texture, perimeter, area, etc.) and a label showing if the tumor was benign or malignant<sup>13</sup>.
- **Features:** Examples include:
  - **Radius Mean:** The average distance from the center to the edge of the tumor.
  - **Texture Mean:** Variation in gray levels in the tumor image.
  - **Perimeter Mean:** The length around the tumor.
  - **Area Mean:** The size of the tumor.
  - **Smoothness Mean:** How smooth the edges of the tumor are.
  - ...and others<sup>13</sup>.

## 2. Data Preprocessing

### Definition:

**Data preprocessing** means cleaning and preparing the data so the computer can use it effectively. This involves:

- **Removing unnecessary columns:** For example, 'id' numbers that do not affect diagnosis.
- **Checking for missing values:** Making sure all measurements are present and correct.
- **Selecting relevant features:** Keeping only the measurements that help in diagnosis.
- **Converting labels:** Changing diagnosis values to "Benign" or "Malignant" for clarity.
- **Splitting the data:** Dividing the data into a training set (to teach the computer) and a test set (to check if it learned well)

## 3. Building the Prediction Model

### Definition:

A **Support Vector Machine (SVM)** is a machine learning algorithm that finds the best way to separate two groups (benign vs. malignant) based on the measurements. It draws a line (or a curve, in more complex cases) that best divides the two types of tumors

- **Training:** The SVM learns from the training data which patterns are more likely to be cancerous.
- **Testing:** We check how well the SVM predicts on new, unseen data.

## 4. Making the Model Explainable

### Definition:

**Explainable Artificial Intelligence (XAI)** refers to methods that help humans understand how and why a computer made a certain prediction

- **SHAP (SHapley Additive exPlanations):** Shows how much each measurement contributed to the final prediction, making the model's decision transparent
- **LIME (Local Interpretable Model-agnostic Explanations):** Builds a simple explanation just for one prediction, helping users understand individual results.

## 5. Creating the Web Application

- **User Interface:** Built using R and Shiny, the app allows users (doctors or patients) to enter the measurements of a tumor.
- **Prediction:** When measurements are entered, the app predicts if the tumor is benign or malignant.
- **Visualization:** The app shows:
  - The prediction result.
  - A **ROC curve** (Receiver Operating Characteristic curve) to show how well the model distinguishes between benign and malignant tumors.
  - A **Confusion Matrix** to display how many predictions were correct or incorrect<sup>12</sup>.

## 6. Evaluating the Model

- **Accuracy:** How often the model is correct.
- **ROC Curve:** A graph that shows the model's ability to separate benign from malignant cases. The closer the curve is to the top-left corner, the better the model.
- **Confusion Matrix:** A table showing true positives, true negatives, false positives, and false negatives, helping us understand where the model might make mistakes

---

## Glossary of Key Terms

Term	Definition
<b>Benign</b>	A non-cancerous tumor that does not spread
<b>Malignant</b>	A cancerous tumor that can invade and spread
<b>Mammogram</b>	An X-ray image of the breast used for cancer detection
<b>Biopsy</b>	Removing a small piece of tissue to check for cancer
<b>Chemotherapy</b>	Treatment using drugs to kill cancer cells

Term	Definition
<b>Support Vector Machine (SVM)</b>	A machine learning model that classifies data into two groups
<b>ROC Curve</b>	A graph showing the performance of a classification model <sup>12</sup> .
<b>Confusion Matrix</b>	A table showing the number of correct and incorrect predictions <sup>12</sup> .
<b>Explainable AI (XAI)</b>	Methods that help humans understand how AI makes decisions
<b>SHAP/LIME</b>	Tools for explaining predictions from machine learning models

---

## Summary

- **Need:** Early, accurate, and understandable breast cancer diagnosis saves lives.
  - **What We Did:** Built an interactive app using SVM to predict if a tumor is benign or malignant, and made the predictions explainable.
  - **How:** Collected and cleaned data, trained an SVM model, added explanation tools, and created a user-friendly web app for predictions and education.
- 

The measurements (features) used in your breast cancer prediction project are specific characteristics of a breast tumor that are obtained from medical images (like those from a biopsy or mammogram). These measurements describe the size, shape, and texture of the tumor. Here are the exact measurements you used, along with simple definitions:

---

## Main Tumor Measurements Used in the Project

Feature Name	What It Means
<b>Radius Mean</b>	The average distance from the center of the tumor to its edge. It gives an idea of how big the tumor is.
<b>Texture Mean</b>	How much the gray levels (brightness) vary in the tumor image. It describes how rough or smooth the tumor looks.
<b>Perimeter Mean</b>	The total length around the edge of the tumor.
<b>Area Mean</b>	The total size (surface area) of the tumor.
<b>Smoothness Mean</b>	How smooth the edges of the tumor are (lower values mean smoother edges).
<b>Compactness Mean</b>	How closely packed the tumor cells are (related to the shape and edge sharpness).
<b>Concavity Mean</b>	How much the edges of the tumor curve inward (concave parts).
<b>Concave Points Mean</b>	The number of places where the edge of the tumor curves inward.
<b>Symmetry Mean</b>	How symmetrical (balanced) the shape of the tumor is.
<b>Fractal Dimension Mean</b>	A measure of the complexity of the tumor's edge (higher values mean a more complex or jagged edge).

These features are entered into the model to help it decide if the tumor is likely to be **benign** (non-cancerous) or **malignant** (cancerous)

---

## Why These Measurements?

- **Size-related features** (radius, perimeter, area) often get larger in malignant (cancerous) tumors.
  - **Shape and edge features** (smoothness, compactness, concavity, concave points) help distinguish between smooth, round benign tumors and irregular, jagged malignant tumors.
  - **Texture and complexity features** (texture, symmetry, fractal dimension) provide extra details that help the model make a more accurate prediction.
- 

## Summary Table

Feature Name	Example Value (Benign)	Example Value (Malignant)
Radius Mean	12.1	17.5
Texture Mean	17.9	21.6
Perimeter Mean	78.1	115.4
Area Mean	462.8	978.4
Smoothness Mean	0.092	0.103
Compactness Mean	0.080	0.145
Concavity Mean	0.046	0.161
Concave Points Mean	0.025	0.088
Symmetry Mean	0.174	0.193
Fractal Dimension Mean	0.063	0.063

*(Values are averages from the dataset for benign and malignant tumors)*

---

### In summary:

The model uses these 10 measurements, which describe the size, shape, and texture of a breast tumor, to predict whether it is benign or malignant

A basic patient will **not** know these tumor measurements on their own. The measurements used in the project (like radius mean, texture mean, perimeter mean, etc.) are **not something a person can feel or see at home**. These are technical features that are calculated by doctors and medical professionals using special medical tests and equipment.

## How are these measurements obtained?

- **Medical Imaging:**  
Doctors use imaging tests like **mammograms** (special X-rays), **ultrasound**, or **MRI** scans to look inside the breast. These machines take detailed pictures of the lump.
- **Biopsy and Lab Analysis:**  
If a lump is found, a **biopsy** may be done. This means a small piece of the lump is removed and

looked at under a microscope by a pathologist (a doctor who studies tissues).

The pathologist uses computers and special software to measure things like the size, shape, and texture of the tumor cells

- **Measurement Details:**

- **Tumor size** is measured in millimeters or centimeters, usually at its widest point
- Other features (like smoothness, symmetry, etc.) are calculated using computer analysis of the images or the tissue sample.

## Who provides these numbers?

- **Doctors and Hospitals:**

After medical imaging or a biopsy, the doctor or hospital will have a report that includes these measurements.

The patient does **not** need to know or calculate these numbers themselves.

## How does this relate to the app?

- The app is designed for use by **doctors, researchers, or trained healthcare staff** who have access to these measurements from medical reports.
- If a patient wants to use the app, they would need to ask their doctor for the detailed measurements from their test results.

---

### In summary:

A basic patient cannot know or measure these values at home. These measurements come from medical tests done by healthcare professionals using specialized equipment and lab analysis. The patient receives these numbers as part of their medical report from their doctor

A basic patient **cannot** enter these measurements into the dashboard by themselves, because:

- The measurements required by the dashboard (like "radius mean," "texture mean," "perimeter mean," etc.) are **not visible or feelable** by a person at home.
- These values are **not symptoms or things you can self-report**; they are **technical numbers** calculated from medical images or lab tests.

## How are these measurements obtained?

- **Medical Imaging (Mammogram, Ultrasound, MRI):**

- Doctors use machines to take detailed pictures of the breast.
- Special computer software analyzes these images and calculates the measurements (like the size, shape, and texture of a lump)

- **Biopsy:**



- If a lump is found, a doctor may take a small tissue sample (biopsy).
- In the lab, experts (pathologists) use microscopes and computers to measure the features of the lump and produce the required numbers

## Who enters the measurements?

- **Doctors or Trained Medical Staff:**

Only healthcare professionals, who have access to the patient's medical imaging and lab reports, can get these numbers and enter them into the dashboard for prediction.

- **Patients:**

If a patient wants to use the dashboard, they would need to:

- Ask their doctor for a copy of their detailed medical report.
- Find the section in the report that lists the measurements (radius mean, texture mean, etc.).
- Enter those exact values into the dashboard.

## Why can't a patient do it alone?

- The measurements are **not based on what you feel or see** (like a lump or pain); they are **scientific values** calculated by machines and experts.
- Without a doctor's report, a patient will not know these numbers.

## What is the dashboard for?

- The dashboard is **mainly for doctors, researchers, or patients who already have a detailed report** from their hospital.
- It helps interpret the results and provides a prediction, but **it does not replace medical tests or professional advice**.

---

### In summary:

A basic patient cannot know or enter these measurements without help from a doctor. These values come from medical tests and reports, not from self-examination or symptoms. For a patient to use the dashboard, they need to get the exact numbers from their doctor or lab report

Certainly! Here are clear, direct answers to your questions based on your project files and previous explanations:

---

## 1. What are the exact measurements used in the project?

The project uses the following 10 measurements (features) to predict if a breast tumor is benign (non-cancerous) or malignant (cancerous):

Feature Name	Definition
<b>Radius Mean</b>	The average distance from the center of the tumor to its edge (size).
<b>Texture Mean</b>	How much the brightness (gray levels) varies in the tumor image (roughness).
<b>Perimeter Mean</b>	The total length around the edge of the tumor.
<b>Area Mean</b>	The total area (surface) of the tumor.
<b>Smoothness Mean</b>	How smooth the edges of the tumor are (lower values = smoother).
<b>Compactness Mean</b>	How closely packed the tumor cells are (relates to edge sharpness).
<b>Concavity Mean</b>	How much the tumor edges curve inward.
<b>Concave Points Mean</b>	The number of inward-curving points on the tumor edge.
<b>Symmetry Mean</b>	How symmetrical the tumor shape is.
<b>Fractal Dimension Mean</b>	A measure of how complex or jagged the tumor edge is.

---

## 2. How does a basic patient know these measurements?

A basic patient **does not** know these measurements on their own. These values are **not symptoms or things you can feel or see at home**. They are technical numbers calculated by doctors or lab professionals using:

- **Medical imaging** (such as mammograms, ultrasound, or MRI): Special machines take detailed pictures of the breast.
  - **Biopsy and lab analysis**: If a lump is found, a small tissue sample is examined under a microscope, and computers/software calculate these measurements.<sup>1</sup>
- 

## 3. If a patient wants to use the dashboard, how do they enter the measurements?

- **A patient cannot measure or guess these values at home.**
- If a patient wants to use the dashboard, they must:
  - **Ask their doctor** for a copy of their detailed medical report after imaging or biopsy.
  - **Find the section in the report** that lists these measurements (radius mean, texture mean, etc.).
  - **Enter those exact values** from the report into the dashboard input fields.<sup>12</sup>

**Without a doctor's report, a patient cannot use the dashboard meaningfully.** The dashboard is primarily intended for doctors, researchers, or patients who already have a detailed medical report.<sup>1</sup>

---

## 4. Why can't a patient do it alone?

- The measurements are **not based on what you feel or see** (like a lump or pain); they are scientific values calculated by machines and experts.
- **Without a doctor's report, a patient will not know these numbers.**
- The dashboard does **not replace medical tests or professional advice**—it helps interpret results from medical reports.<sup>12</sup>

---

## 5. Summary Table of Features

Feature Name	Example Value (Benign)	Example Value (Malignant)
Radius Mean	12.1	17.5
Texture Mean	17.9	21.6
Perimeter Mean	78.1	115.4
Area Mean	462.8	978.4
Smoothness Mean	0.092	0.103
Compactness Mean	0.080	0.145
Concavity Mean	0.046	0.161
Concave Points Mean	0.025	0.088
Symmetry Mean	0.174	0.193
Fractal Dimension Mean	0.063	0.063

*(Values are averages from the dataset for benign and malignant tumors.)*

---

## 6. In Summary

- **Patients cannot know or enter these measurements without a doctor's report.**
- **Doctors or trained staff** use medical test results to get these numbers and enter them into the dashboard for prediction.
- The dashboard is a tool to help interpret medical data, **not a replacement for professional medical testing or advice.**<sup>12</sup>

---

If you need sample viva questions, definitions, or further clarifications, let me know!

You're asking:

Given the values in the table (for example, "Radius Mean: 12.1 for Benign, 17.5 for Malignant"), and since these come from the dataset, **how do we know which value is benign and which is malignant? In other words, how do we know what range or value means benign or malignant?**

---

# How Do We Know Which Value Is Benign or Malignant?

## 1. The Dataset Labels Each Sample

- In the original dataset (`breast-cancer-wisconsin-data.csv`), each row (patient case) has a **diagnosis label**:
  - 'B' for Benign
  - 'M' for Malignant
- Each row also has the actual measurements (`radius_mean`, `texture_mean`, etc.) for that case<sup>3</sup>.

## 2. Statistical Analysis of Feature Values

- By looking at all the benign cases (rows labeled 'B'), we can calculate the **average (mean) value** for each measurement for benign tumors.
- Similarly, for all malignant cases (rows labeled 'M'), we calculate the average for each measurement for malignant tumors.
- The table you provided is a summary: it shows the **average value of each feature for benign and malignant tumors**<sup>8</sup>.

### Example:

- If you look at all benign cases, the average (mean) `radius_mean` is about 12.1.
- For malignant cases, the average `radius_mean` is about 17.5.

## 3. Overlap and Ranges

- **There is overlap:**  
Not every benign tumor will have a `radius_mean` less than 12.1, and not every malignant tumor will have a value above 17.5. These are just averages.
- **Ranges:**  
The actual values for benign and malignant tumors form distributions (ranges), and the model learns from these distributions to find patterns<sup>8</sup>.

## 4. How the Model Learns

- The machine learning model (SVM) is trained on the dataset, using the actual labels ('B' or 'M') and the feature values.

- The model **learns the patterns**: for example, higher radius\_mean, perimeter\_mean, and area\_mean are more commonly seen in malignant tumors, but it uses all features together to make a decision

## 5. No Single Cutoff

- There is **no single cutoff value** for each measurement that always means benign or malignant. Instead, the model looks at **all the measurements together** to make its prediction, based on how similar the new case is to the patterns it learned from the labeled data.

---

### Summary Table (for clarity)

Feature Name	Average for Benign	Average for Malignant
Radius Mean	12.1	17.5
Texture Mean	17.9	21.6
Perimeter Mean	78.1	115.4
Area Mean	462.8	978.4
Smoothness Mean	0.092	0.103
Compactness Mean	0.080	0.145
Concavity Mean	0.046	0.161
Concave Points Mean	0.025	0.088
Symmetry Mean	0.174	0.193
Fractal Dimension Mean	0.063	0.063

These numbers are **averages** from the dataset. The model uses the full range of values and the patterns in the data to make predictions

---

### In Simple Terms

- The dataset tells us which cases are benign and which are malignant.
- By looking at the measurements for each group, we see that, on average, malignant tumors have higher values for most features.
- The model learns from these patterns, but real-life cases can overlap, so the model uses all features together to decide.
- **\*\*You don't decide "benign" or "malignant" just by looking at one value—the model considers all the measurements and the patterns it learned from the labeled data**

---

#### In summary:

We know which value is benign or malignant because the dataset tells us (with the 'diagnosis' label), and the averages you see are calculated by grouping all benign or malignant cases and finding the mean

for each measurement. The machine learning model uses these patterns—not just single values—to predict new cases.

There is **no single formula** that can tell you, just from one measurement, whether a tumor is benign or malignant. Instead, the prediction is made by a **machine learning model** (like SVM) that looks at **all the measurements together** and learns patterns from the labeled dataset.

## How Does the Model Decide?

- Each row in your dataset has a set of measurements (features) and a label: **Benign (B)** or **Malignant (M)**<sup>3</sup>.
- The model (SVM) is trained on this data. It learns from **many examples** what combinations of measurements are likely to be benign or malignant<sup>2</sup>.
- When you enter new measurements, the model uses what it has learned to predict the label.

## Are There Any Mathematical Formulas?

For some individual features, researchers have used formulas to describe tumor shape or complexity, such as:

- **Compactness:**

$$\text{Compactness} = \frac{(\text{Perimeter})^2}{\text{Area}}$$

Higher compactness usually means a more irregular shape, which is more common in malignant tumors<sup>67</sup>.

- **Radius Mean:**

The average distance from the center to the edge of the tumor<sup>69</sup>.

- **Other shape features** (like concave points, symmetry, fractal dimension) are calculated using image analysis and mathematical definitions, but **none of these alone can classify a tumor as benign or malignant**.

## Why No Simple Cutoff or Formula?

- **Overlap:** Benign and malignant tumors can have overlapping values for each feature. For example, a malignant tumor might have a radius mean of 15, but so might a benign one.

- **Combinations Matter:** The model looks at **all the features together**. For example, a large radius plus high compactness plus high concavity might be more likely malignant, but each feature alone is not enough.
- **The SVM Model:** SVM creates a complex boundary (not a simple formula) in multi-dimensional space to separate benign from malignant cases based on all features<sup>2</sup>.

## How Does the Model Learn?

- The model uses the **training data** (with known labels) to find patterns.
- It then applies these patterns to new data to predict the label.

## In Summary

- **There is no single mathematical formula or cutoff for any one measurement to decide benign/malignant.**
- The **machine learning model** uses all the measurements together and learns from the labeled data.
- Some features (like compactness) have mathematical formulas, but the final decision is made by the trained model, **not a rule or formula**<sup>679</sup>.

---

**If you need to classify a tumor, you must use the trained model, not a formula. The model does the complex math for you!**

Here's a clear explanation of **how we train the model** for breast cancer prediction using SVM, step by step, with references to your project and standard practice:

---

## Step-by-Step: How We Train the Model

### 1. Data Preparation

- **Collect the Data:**  
We use the Breast Cancer Wisconsin Diagnostic dataset, which contains measurements (features) of tumors and labels (benign or malignant)<sup>1235</sup>.

- **Clean and Explore:**

We check for missing values, outliers, and understand the distribution of the data[23](#).

## 2. Feature Selection and Scaling

- **Select Features:**

We select the relevant measurements (like radius mean, texture mean, etc.) as input features (X)[12](#).



- **Label Encoding:**

The diagnosis column is converted to numbers (e.g., benign = 0, malignant = 1)[2](#).

- **Feature Scaling:**

Since measurements have different units and scales, we scale them (using methods like MinMaxScaler or StandardScaler) so that all features contribute equally[23](#).

## 3. Splitting the Data

- **Training and Test Sets:**

The dataset is split into a **training set** (usually 70–80%) and a **test set** (20–30%).

- The **training set** is used to teach the model.
- The **test set** is used to check how well the model works on new, unseen data[23](#).

## 4. Model Initialization

- **Choose SVM Algorithm:**

We use Support Vector Machine (SVM) for classification.

- **Set Hyperparameters:**

We choose a kernel (linear or RBF), and set parameters like C (regularization) and gamma (for RBF kernel)[35](#).

## 5. Model Training

- **Fit the Model:**

We use the training data to let the SVM model learn the patterns that separate benign from malignant tumors.

This is done using the `.fit()` method in libraries like scikit-learn:

```
python
```



```
from sklearn.svm import SVC
svc_model = SVC(kernel='rbf', C=1.0, gamma='auto')

svc_model.fit(X_train, y_train)
```

The model finds the best boundary (decision surface) between the two classes[235](#).

## 6. Model Evaluation

- **Predict on Test Data:**

We use the trained model to predict the labels of the test set.



- **Measure Performance:**

We use metrics like accuracy, precision, recall, F1-score, confusion matrix, and ROC curve to see how well the model is doing[235](#).

## 7. Model Tuning (Optimization)

- **Cross-Validation:**

We use k-fold cross-validation to make sure the model works well on different subsets of the data[3](#).

- **Hyperparameter Tuning:**

We try different values of kernel, C, and gamma to find the best combination (using grid search)[35](#).

---

## Summary Table

Step	What Happens
Data Preparation	Clean, explore, and preprocess the data
Feature Scaling	Normalize all features to the same scale
Data Splitting	Split into training and test sets
Model Initialization	Choose SVM, set kernel and parameters
Model Training	Fit SVM to training data (learn patterns)
Model Evaluation	Test predictions, calculate accuracy, confusion matrix, etc.
Model Tuning	Use cross-validation and grid search to improve performance

---

## References to Your Sources

- [Kaggle SVM Breast Cancer Example][1](#)
- [Towards Data Science SVM Case Study][2](#)
- [African Journal of Biomedical Research: SVM Training Process][3](#)
- [JATIT: SVM Tuning and Optimization][5](#)

---

### In short:

We train the model by giving it many examples of tumor measurements and their correct labels, letting it learn the patterns, and then testing it on new data to make sure it predicts accurately. We also tune and optimize the model for best results.

---

If you want, I can provide the actual Python code for this process as well!

Here's a clear, section-by-section explanation of your **app.R** code for the Breast Cancer Detection App, written in R with Shiny and SVM:

---

## 1. Loading Libraries

```
r
library(tidyverse) # For data manipulation
library(caret)     # For machine learning utilities (like data splitting)
library(shiny)     # For building interactive web apps
library(shinydashboard) # For dashboard-style UI
library(ggplot2)   # For plotting
library(DT)        # For interactive tables
library(pROC)      # For ROC curve plotting
library(e1071)     # For SVM model
```

**Purpose:**

These libraries provide all the tools needed for data processing, modeling, visualization, and the web interface.

---

## 2. Loading and Preparing the Data

r

```
data <- read.csv("E:/DOWNLOADS/breast-cancer-wisconsin-data.csv")
data <- data %>% select(-id)

data <- data %>% select(diagnosis, radius_mean, texture_mean, perimeter_mean, area_mean,
                        smoothness_mean, compactness_mean, concavity_mean,
                        concave.points_mean, symmetry_mean, fractal_dimension_mean)

data$diagnosis <- factor(data$diagnosis, levels = c("B", "M"), labels = c("Benign", "Malignant"))

if (sum(is.na(data)) > 0) {
  stop("Data contains missing values.")
}
```

**Purpose:**

- Loads the dataset.
  - Removes the unnecessary `id` column.
  - Selects only the relevant features for prediction.
  - Converts the `diagnosis` column to a factor with labels "Benign" and "Malignant".
  - Checks for missing values and stops if any are found.
- 

## 3. Splitting Data into Training and Testing Sets

r

```
set.seed(123)
index <- createDataPartition(data$diagnosis, p = 0.7, list = FALSE)
```

```
train_data <- data[index, ]
```

```
test_data <- data[-index, ]
```

**Purpose:**

- Randomly splits the data into 70% for training and 30% for testing.
- Ensures reproducibility with `set . seed(123)`.

---

## 4. Training the SVM Model

r

```
svm_model <- svm(diagnosis ~ ., data = train_data, kernel = "radial", cost = 1, gamma = 0.1, probability = TRUE)
```

**Purpose:**

- Trains a Support Vector Machine (SVM) classifier using the training data.
- Uses a radial basis function (RBF) kernel, which is good for non-linear data.
- Enables probability estimates for predictions.

---

## 5. User Interface (UI) Definition

r

```
ui <- dashboardPage(
  dashboardHeader(title = "Breast Cancer Detection", titleWidth = 250),
```

```
  dashboardSidebar(
```

```
    width = 250,
```

```
    sidebarMenu(
```

```
menuItem("Dashboard", tabName = "dashboard", icon = icon("chart-line")),

menuItem("About", tabName = "about", icon = icon("info-circle")),

menuItem("Help", tabName = "help", icon = icon("question-circle"))

)

),

dashboardBody(

# Custom styles and tab definitions (Dashboard, About, Help)

)

)
```

### Purpose:

- Sets up the structure of the app with a header, sidebar, and main body.
- Includes three tabs: Dashboard (main tool), About (info), Help (instructions).

---

## Dashboard Tab

- **Input Boxes:**  
Users manually enter the 10 tumor measurements (numericInput for each).
  - **Predict Button:**  
When clicked, triggers the prediction.
  - **Output:**
    - Shows the prediction result (benign or malignant).
    - Displays the ROC curve (model performance).
    - Shows the confusion matrix (prediction accuracy).
-

## 6. Server Logic

r

```
server <- function(input, output, session) {  
  
  # Prediction result  
  
  output$prediction_result <- renderText({  
  
    req(input$predict_btn) # Wait for Predict button  
  
    new_data <- data.frame(  
      radius_mean = input$radius_mean,  
      texture_mean = input$texture_mean,  
      perimeter_mean = input$perimeter_mean,  
      area_mean = input$area_mean,  
      smoothness_mean = input$smoothness_mean,  
      compactness_mean = input$compactness_mean,  
      concavity_mean = input$concavity_mean,  
      concave.points_mean = input$concave.points_mean,  
      symmetry_mean = input$symmetry_mean,  
      fractal_dimension_mean = input$fractal_dimension_mean  
    )  
  
    pred <- predict(svm_model, new_data)  
  
    paste("The predicted diagnosis is:", pred)
```

```

    })

    # ROC Curve

    output$rocPlot <- renderPlot({

    prob <- attr(predict(svm_model, test_data, probability = TRUE), "probabilities")[, "Malignant"]

    roc_curve <- roc(test_data$diagnosis, prob, levels = c("Benign", "Malignant"))

    plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)

    })

    # Confusion Matrix

    output$confMatrix <- renderDT({

    predictions <- predict(svm_model, test_data)

    confusionMatrix(predictions, test_data$diagnosis)$table

    })
  }
}

```

## Purpose:

- **Prediction:**  
When the user enters values and clicks Predict, the app creates a new data frame, uses the trained SVM model to predict, and displays the result.
- **ROC Curve:**  
Plots the ROC curve for the test data, showing the model's ability to distinguish benign from malignant.

- **Confusion Matrix:**  
Shows a table of true/false positives/negatives for model performance evaluation.
- 

## 7. Running the App

r

```
shinyApp(ui = ui, server = server)
```

### Purpose:

- Launches the Shiny web application with the defined UI and server logic.
- 

## Summary Table

Section	What It Does
Load Libraries	Import necessary R packages
Load & Prepare Data	Read CSV, select features, label encoding, check for missing values
Split Data	Create training and test sets
Train Model	Train SVM classifier on training data
UI Definition	Build the web interface (inputs, tabs, outputs)
Server Logic	Handle user input, make predictions, plot ROC, show confusion matrix
Run App	Start the Shiny app

---

### In summary:

This app loads breast cancer data, trains an SVM model, and provides an interactive dashboard for users to input tumor measurements and get a prediction (benign or malignant), along with performance metrics like ROC and confusion matrix—all in a user-friendly web interface.

### More Detailed explanation

Here is a **detailed, step-by-step explanation** of your app . R code for the Breast Cancer Detection App, covering **every detail** from data loading to user interaction, model training, prediction, and output visualization.

---

## 1. Loading Libraries

r



```

library(tidyverse) # Data manipulation and wrangling (dplyr, ggplot2, etc.)
library(caret)     # Machine learning utilities (data splitting, evaluation)

library(shiny)     # Web app framework for R

library(shinydashboard) # Dashboard-style UI for Shiny

library(ggplot2)   # Data visualization

library(DT)        # Interactive tables

library(pROC)      # ROC curve plotting

library(e1071)     # SVM model implementation

```

### Purpose:

These libraries provide all the tools for data processing, visualization, machine learning, and building the interactive web app<sup>45</sup>.

## 2. Data Loading and Preparation

```

r
data <- read.csv("E:/DOWNLOADS/breast-cancer-wisconsin-data.csv")
data <- data %>% select(-id)

data <- data %>% select(diagnosis, radius_mean, texture_mean, perimeter_mean, area_mean,
                        smoothness_mean, compactness_mean, concavity_mean,
                        concave.points_mean, symmetry_mean, fractal_dimension_mean)

data$diagnosis <- factor(data$diagnosis, levels = c("B", "M"), labels = c("Benign", "Malignant"))

if (sum(is.na(data)) > 0) {
  stop("Data contains missing values.")
}

```

#### Details:

- **Read CSV:** Loads the breast cancer dataset.
  - **Remove 'id':** The 'id' column is just an identifier, not useful for prediction.
  - **Select Features:** Keeps only the diagnosis and the 10 most important features (mean values of radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension)[5](#).
  - **Convert Diagnosis:** Changes 'B'/'M' to "Benign"/"Malignant" for clarity.
  - **Check for Missing Values:** If any are found, stops the app to prevent errors.
- 

### 3. Data Splitting

r

```
set.seed(123)
index <- createDataPartition(data$diagnosis, p = 0.7, list = FALSE)

train_data <- data[index, ]

test_data <- data[-index, ]
```

#### Details:

- **Set Seed:** Ensures reproducibility (same split every time).
  - **Partition Data:** Splits the data into 70% for training and 30% for testing using stratified sampling (keeps the ratio of benign/malignant similar in both sets)[45](#).
- 

### 4. Model Training (SVM)

r

```
svm_model <- svm(diagnosis ~ ., data = train_data, kernel = "radial", cost = 1, gamma = 0.1, probability = TRUE)
```

#### Details:

- **SVM:** Trains a Support Vector Machine classifier using the training data.
- **Formula:** `diagnosis ~ .` means use all other columns to predict diagnosis.
- **Kernel:** 'radial' (RBF) kernel enables non-linear decision boundaries, which is important for complex medical data<sup>4</sup>.
- **Cost and Gamma:** Control the trade-off between margin width and misclassification (cost) and the influence of single data points (gamma).
- **Probability:** Enables probability predictions for plotting ROC curves.

---

## 5. User Interface (UI) Definition

r

```

ui <- dashboardPage(
  dashboardHeader(title = "Breast Cancer Detection", titleWidth = 250),
  dashboardSidebar(
    sidebarMenu(
      menuItem("Dashboard", tabName = "dashboard", icon = icon("chart-line")),
      menuItem("About", tabName = "about", icon = icon("info-circle")),
      menuItem("Help", tabName = "help", icon = icon("question-circle"))
    )
  ),
  dashboardBody(
    # Custom CSS for theme and layout
    tags$head(tags$style(HTML('...'))),

```

```

    tabItems(
      # Dashboard Tab
      tabItem(tabName = "dashboard",
        fluidRow(
          box(title = "Input Features", width = 6, solidHeader = TRUE,
            numericInput("radius_mean", ...),
            numericInput("texture_mean", ...),
            # ... other numericInputs for all 10 features
            actionButton("predict_btn", "Predict", ...),
            h4("Prediction Result:", ...),
            textOutput("prediction_result", inline = TRUE),
            textOutput("prediction_status")
          ),
          box(plotOutput("rocPlot"), width = 6),
          box(DTOutput("confMatrix"), width = 6)
        )
      ),
      # About Tab
      tabItem(tabName = "about", ...),
      # Help Tab

```

```
tabItem(tabName = "help", ...)
)
)
)
```

## Details:

- **Dashboard Layout:** Sidebar for navigation, main area for input/output.
- **Input Features:** User enters 10 measurements (mean values) for a tumor.
- **Predict Button:** User clicks to get a prediction.
- **Outputs:** Prediction result, ROC curve, and confusion matrix.
- **About/Help Tabs:** Provide background info and instructions for users.

---

## 6. Server Logic

r

```
server <- function(input, output, session) {  
  
  # Prediction  
  
  output$prediction_result <- renderText({  
  
    req(input$predict_btn)  
  
    new_data <- data.frame(  
  
      radius_mean = input$radius_mean,  
  
      texture_mean = input$texture_mean,  
  
      perimeter_mean = input$perimeter_mean,
```

```

        area_mean = input$area_mean,
        smoothness_mean = input$smoothness_mean,
        compactness_mean = input$compactness_mean,
        concavity_mean = input$concavity_mean,
        concave.points_mean = input$concave.points_mean,
        symmetry_mean = input$symmetry_mean,
        fractal_dimension_mean = input$fractal_dimension_mean
    )

    pred <- predict(svm_model, new_data)

    paste("The predicted diagnosis is:", pred)

})

# ROC Curve

output$rocPlot <- renderPlot({

    prob <- attr(predict(svm_model, test_data, probability = TRUE), "probabilities")[, "Malignant"]

    roc_curve <- roc(test_data$diagnosis, prob, levels = c("Benign", "Malignant"))

    plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)

})

# Confusion Matrix

```

```

output$confMatrix <- renderDT({
  predictions <- predict(svm_model, test_data)
  confusionMatrix(predictions, test_data$diagnosis)$table
})
}

```

## Details:

- **Prediction:**
  - Waits for the Predict button.
  - Collects user inputs into a data frame.
  - Uses the trained SVM model to predict diagnosis.
  - Displays the result as "Benign" or "Malignant."
- **ROC Curve:**
  - Gets probability predictions for the test set.
  - Plots the ROC curve, showing the model's ability to distinguish between classes.
- **Confusion Matrix:**
  - Predicts diagnoses for the test set.
  - Shows a table of true positives, true negatives, false positives, and false negatives, helping users understand model accuracy.

---

## 7. Running the App

r

```
shinyApp(ui = ui, server = server)
```

## Details:

- Launches the Shiny app with the defined UI and server logic, making it interactive and accessible via a web browser.
- 

## End-to-End Workflow

1. **Data is loaded and cleaned** (irrelevant columns removed, missing values checked).
  2. **Data is split** into training and test sets.
  3. **SVM model is trained** on the training data using 10 key features.
  4. **Web interface is built** for user interaction.
  5. **User enters measurements** and clicks "Predict."
  6. **App predicts diagnosis** using the trained SVM model.
  7. **App displays:**
    - The prediction (benign/malignant)
    - ROC curve (model performance)
    - Confusion matrix (detailed accuracy breakdown)
  8. **About and Help tabs** provide background and instructions.
- 

## Why This Approach?

- **SVM is robust and effective** for high-dimensional medical data, especially with a non-linear kernel like RBF<sup>45</sup>.
  - **Interactive dashboard** makes it accessible for clinicians and researchers.
  - **Performance metrics (ROC, confusion matrix)** provide transparency and trust in the model's predictions.
  - **Feature selection** ensures only the most relevant measurements are used, reducing noise and improving interpretability<sup>3</sup>.
-



## References to Best Practices

- Data cleaning, feature selection, and model evaluation are standard and recommended in machine learning for medical diagnostics<sup>[345](#)</sup>.
  - SVM with RBF kernel is widely used for breast cancer prediction due to its ability to handle non-linear boundaries and high-dimensional data<sup>[245](#)</sup>.
  - Visualizations like ROC and confusion matrix help users understand and trust the model's performance<sup>[34](#)</sup>.
- 

### In summary:

Your app loads, cleans, and splits the breast cancer data, trains an SVM model, and provides an interactive dashboard for users to enter tumor measurements and receive a prediction, along with detailed model performance visualizations and educational content. This ensures both accuracy and usability for real-world clinical or research settings.

Certainly! Here's a clear explanation of **True Positive (TP)**, **True Negative (TN)**, **False Positive (FP)**, and **False Negative (FN)** in the context of your breast cancer detection project:

---

## Definitions in Breast Cancer Prediction Context

Imagine your model is predicting whether a tumor is **Malignant (cancerous)** or **Benign (not cancerous)**.

### 1. True Positive (TP)

- **What it means:**  
The model predicted the tumor is **Malignant** (cancer), and it actually is Malignant.
- **Example:**  
The model says "Malignant," and the real diagnosis is also "Malignant."

### 2. True Negative (TN)

- **What it means:**  
The model predicted the tumor is **Benign** (not cancer), and it actually is Benign.

- **Example:**  
The model says “Benign,” and the real diagnosis is also “Benign.”

### 3. False Positive (FP)

- **What it means:**  
The model predicted the tumor is **Malignant** (cancer), but it is actually Benign.
- **Example:**  
The model says “Malignant,” but the real diagnosis is “Benign.”  
(This is also called a “Type I error.”)

### 4. False Negative (FN)

- **What it means:**  
The model predicted the tumor is **Benign** (not cancer), but it is actually Malignant.
- **Example:**  
The model says “Benign,” but the real diagnosis is “Malignant.”  
(This is also called a “Type II error.”)

---

## Summary Table

	Actual Benign	Actual Malignant
Predicted Benign	True Negative (TN)	False Negative (FN)
Predicted Malignant	False Positive (FP)	True Positive (TP)

---

## Why Are These Important?

- **True Positives (TP):**  
Correctly identifying cancer—very important for treatment.
- **True Negatives (TN):**  
Correctly identifying non-cancer—avoids unnecessary worry or treatment.
- **False Positives (FP):**  
Incorrectly saying someone has cancer—may cause stress and unnecessary tests.
- **False Negatives (FN):**  
Missing a cancer diagnosis—very dangerous, as the patient may not get needed treatment.

---

## In Your App (Confusion Matrix Output):

When you look at the confusion matrix in your dashboard, you can see:

- How many predictions were correct (TP and TN)
- How many were incorrect (FP and FN)

This helps you (and doctors) understand how reliable the model is and where it might make mistakes.

---

### In summary:

- **TP:** Model correctly predicts cancer.
- **TN:** Model correctly predicts no cancer.
- **FP:** Model wrongly predicts cancer.
- **FN:** Model wrongly predicts no cancer.

Understanding these helps evaluate and trust your breast cancer prediction model!

---

Absolutely! Here are **comprehensive viva questions and answers** covering the entire breast cancer detection project, from data to model to app and interpretation:

---

## Viva Questions and Answers

---

### 1. What is the main goal of your project?

#### Answer:

The main goal is to build an interactive tool that predicts whether a breast tumor is benign (non-cancerous) or malignant (cancerous) using measurements from medical tests, and to provide clear explanations and performance metrics to help doctors make informed decisions.

---

## 2. What dataset did you use, and what are its key features?

**Answer:**

We used the Wisconsin Breast Cancer Diagnostic dataset. It contains measurements of breast tumors such as radius mean, texture mean, perimeter mean, area mean, smoothness mean, compactness mean, concavity mean, concave points mean, symmetry mean, and fractal dimension mean, along with a diagnosis label (benign or malignant).

---

## 3. Why do we need early detection of breast cancer?

**Answer:**

Early detection increases the chances of successful treatment and survival. It helps catch cancer before it spreads, making treatment easier and more effective.

---

## 4. What is a Support Vector Machine (SVM), and why did you use it?

**Answer:**

SVM is a machine learning algorithm used for classification. It finds the best boundary to separate two classes (benign and malignant) based on the input features. We used SVM because it is effective for high-dimensional, non-linear data like medical measurements.

---

## 5. What preprocessing steps did you perform?

**Answer:**

- Removed unnecessary columns (like 'id').
  - Selected relevant features.
  - Converted diagnosis labels to factors ("Benign" and "Malignant").
  - Checked for and handled missing values.
  - Split the data into training and testing sets.
-

## 6. How did you split your data, and why?

**Answer:**

We split the data into 70% for training and 30% for testing. This allows the model to learn from one part of the data and be evaluated on unseen data to check its real-world performance.

---

## 7. What is feature scaling, and why is it important?

**Answer:**

Feature scaling adjusts the range of the input features so that no single feature dominates the model due to its scale. It is important for SVM because it relies on distance calculations.

---

## 8. How does your model make a prediction?

**Answer:**

The model takes the 10 tumor measurements as input, applies the patterns it learned during training, and predicts whether the tumor is benign or malignant.

---

## 9. What is the confusion matrix, and what do TP, TN, FP, and FN mean?

**Answer:**

A confusion matrix is a table that shows the number of correct and incorrect predictions:

- **True Positive (TP):** Model correctly predicts malignant.
  - **True Negative (TN):** Model correctly predicts benign.
  - **False Positive (FP):** Model predicts malignant, but it is benign.
  - **False Negative (FN):** Model predicts benign, but it is malignant.
-

## 10. What is an ROC curve and what does AUC mean?

**Answer:**

An ROC curve plots the true positive rate against the false positive rate at different thresholds. AUC (Area Under Curve) measures the overall ability of the model to distinguish between classes; the closer to 1, the better.

---

## 11. How do you ensure the model is explainable?

**Answer:**

We use tools like SHAP and LIME to show which features contributed most to each prediction, helping doctors understand the model's reasoning.

---

## 12. Can a patient use your dashboard at home?

**Answer:**

No. The required measurements are technical values obtained from medical imaging and lab analysis. Only a doctor or someone with a detailed medical report can enter these values.

---

## 13. What happens if a patient enters random or incorrect values?

**Answer:**

The model will still give a prediction, but it will not be meaningful or reliable. The input values must come from real medical tests.

---

## 14. What are the risks of false positives and false negatives in this context?

**Answer:**

- **False Positive:** May cause unnecessary stress and further tests for a patient who does not have cancer.

- **False Negative:** More dangerous, as a patient with cancer may not get timely treatment.
- 

## 15. How did you evaluate your model's performance?

### Answer:

We used metrics like accuracy, confusion matrix, and ROC-AUC score on the test data to measure how well the model predicts new cases.

---

## 16. What is the main advantage of using a dashboard for this model?

### Answer:

It provides an easy-to-use interface for doctors to input measurements, get instant predictions, and see visual explanations and performance metrics.

---

## 17. What are the limitations of your project?

### Answer:

- Requires technical measurements from medical tests.
  - Cannot be used by patients without a doctor's report.
  - Performance depends on the quality and diversity of the dataset.
- 

## 18. How could you improve this project in the future?

### Answer:

- Integrate more features (like genetic or lifestyle data).
- Use larger, more diverse datasets.
- Add more advanced explainability tools.
- Connect directly with hospital databases for automatic input.

---

## 19. Why is explainability important in medical AI?

### Answer:

Doctors need to trust and understand AI predictions, especially for critical decisions like cancer diagnosis. Explainability builds trust and helps in clinical decision-making.

---

## 20. What is the role of the confusion matrix in your dashboard?

### Answer:

It helps users see how many predictions were correct or incorrect, providing transparency about the model's strengths and weaknesses.

---

If you need more specific questions or deeper technical ones, let me know!

---

The **accuracy formula** in classification (including your breast cancer prediction project) measures how many predictions your model got correct out of all predictions made. It is calculated using the values from the confusion matrix: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)[3569](#).

---

## Accuracy Formula

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}$$
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

Or, using the confusion matrix terms:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

- **TP (True Positive):** Model correctly predicts malignant.
- **TN (True Negative):** Model correctly predicts benign.



- **FP (False Positive):** Model predicts malignant, but it is benign.
- **FN (False Negative):** Model predicts benign, but it is malignant.

---

## Example

If your model made 100 predictions:

- 50 were TP
- 40 were TN
- 5 were FP
- 5 were FN

Accuracy =  $\frac{50 + 40}{50 + 40 + 5 + 5} = \frac{90}{100} = 0.9$  or 90%  
 $\text{Accuracy} = \frac{50 + 40 + 5 + 5}{100} = 0.9$  or 90%

---

### In summary:

Accuracy tells you the proportion of total predictions that were correct

\*\*\*\*\*

To **differentiate between benign and malignant breast tumors**, clinicians and computer models use a combination of **quantitative measurements** (like size, shape, and imaging values) and **qualitative features** (like regularity of the tumor edge). Here's how the values are interpreted according to your data and the scientific literature:

---

## 1. Imaging/Quantitative Thresholds

### A. ADC (Apparent Diffusion Coefficient) Values from MRI

- **Malignant lesions:**  
Typically have **lower ADC values** (indicating higher cellularity and restricted diffusion).

- Mean ADC for malignant:  $\sim 0.68\text{--}0.94 \times 10^{-3} \text{ mm}^2/\text{s}$ [124](#)
  - **Benign lesions:**  
Have **higher ADC values** (less cellular, more water diffusion).
    - Mean ADC for benign:  $\sim 1.1\text{--}1.7 \times 10^{-3} \text{ mm}^2/\text{s}$ [124](#)
  - **Threshold (cut-off) values:**
    - A commonly used threshold to separate benign and malignant is  $\sim 1.0\text{--}1.3 \times 10^{-3} \text{ mm}^2/\text{s}$ .
    - Lesions with ADC **below the threshold** are more likely malignant; those **above** are more likely benign[24](#).
- 

## B. Hemoglobin Concentration (Optical Tomography)

- **Malignant lesions:**  
Higher mean and maximum hemoglobin concentration.
    - Mean max for malignant:  $\sim 122 \text{ }\mu\text{mol/L}$
  - **Benign lesions:**  
Lower hemoglobin concentration.
    - Mean max for benign:  $\sim 55\text{--}66 \text{ }\mu\text{mol/L}$
  - **Threshold:**
    - A cut-off of  $95 \text{ }\mu\text{mol/L}$  for max hemoglobin concentration can help separate malignant from benign[5](#).
- 

## C. Shape and Morphological Features (from Ultrasound or Imaging)

- **Benign tumors:**
  - Usually **well-defined, round or oval** in shape, and have **smooth** contours[3](#).
- **Malignant tumors:**

- Often **poorly defined, irregular**, and have **lobulated or spiculated** (spiky) edges<sup>3</sup>.

---

## 2. Features Used in Your Project

Your model uses features like:

- **Radius Mean, Texture Mean, Perimeter Mean, Area Mean, Smoothness Mean, Compactness Mean, Concavity Mean, Concave Points Mean, Symmetry Mean, Fractal Dimension Mean**

**Typical values (from your dataset):**

Feature	Benign (Average)	Malignant (Average)
Radius Mean	12.1	17.5
Texture Mean	17.9	21.6
Perimeter Mean	78.1	115.4
Area Mean	462.8	978.4
Smoothness Mean	0.092	0.103
Compactness Mean	0.080	0.145
Concavity Mean	0.046	0.161
Concave Points Mean	0.025	0.088
Symmetry Mean	0.174	0.193
Fractal Dimension Mean	0.063	0.063

- **Malignant tumors** tend to have **higher values** for most of these features (larger, more irregular, less smooth).
- **Benign tumors** tend to have **lower values** and more regular shapes.

**Note:** There is **no single cut-off value** for these features; the model learns patterns from all features together to make a prediction.

---

## 3. Summary Table of Key Differentiators

Measurement/Feature	Benign Tumor	Malignant Tumor	Typical Threshold
ADC Value (MRI, $\times 10^{-3}$ mm <sup>2</sup> /s)	1.1–1.7	0.68–0.94	1.0–1.3
Hemoglobin (μmol/L)	38–66 (mean/average)	88–122 (mean/average)	95 (max)
Shape	Round, smooth, well-defined	Irregular, spiculated, lobular	N/A

Measurement/Feature	Benign Tumor	Malignant Tumor	Typical Threshold
Size/Texture (dataset)	Lower	Higher	N/A (model-based)

---

## Key Points

- **No single value** guarantees benign or malignant—there is overlap, but these are the general trends.
  - **Models use all features together** (not just one) to make the most accurate prediction.
  - **Thresholds (e.g., ADC of  $1.0\text{--}1.3 \times 10^{-3} \text{ mm}^2/\text{s}$ )** are used in imaging to help radiologists decide, but are not absolute.
- 

### In summary:

Malignant tumors generally have lower ADC values, higher hemoglobin concentrations, and more irregular and larger shapes compared to benign tumors. In your project, the machine learning model uses all available features and their patterns, not just a single cut-off, to differentiate between benign and malignant cases

Yes, **before entering the values into the dashboard**, doctors must perform specific **lab and imaging tests** to obtain the tumor measurements. Here’s a detailed list of the main tests and how they provide the values used in your breast cancer prediction project:

---

## Main Lab and Imaging Tests to Obtain Tumor Measurements

### 1. Mammogram

- **What it is:**  
An X-ray of the breast that helps detect abnormal areas or lumps.
- **What it provides:**
  - Size, shape, and location of a lump.
  - Digital mammography can provide measurements like area, perimeter, and sometimes texture<sup>[127](#)</sup>.

---

## 2. Breast Ultrasound

- **What it is:**  
Uses high-frequency sound waves to create images of breast tissue.
- **What it provides:**
  - Helps distinguish between solid tumors and fluid-filled cysts.
  - Measures the size and sometimes the texture and edge characteristics of a lump

---

## 3. Breast MRI (Magnetic Resonance Imaging)

- **What it is:**  
Uses magnetic fields and radio waves to create detailed images of the breast.
- **What it provides:**
  - Very accurate measurements of tumor size, shape, and sometimes internal features (like smoothness, symmetry, and complexity).
  - Especially useful for dense breast tissue or when mammogram results are unclear

---

## 4. Biopsy (Tissue Sampling)

- **What it is:**  
Removal of a small sample of breast tissue for laboratory analysis.
  - Types: Core needle biopsy, fine needle aspiration, punch biopsy, vacuum-assisted biopsy
- **What it provides:**
  - Pathologists examine the tissue under a microscope and use computers to measure:
    - Cell shape, size, and arrangement
    - Edge smoothness, compactness, concavity, and other features
  - These measurements are often digitized and can be used as input for models

---

## 5. Additional Imaging and Tests (if needed)

- **CT Scan, PET/CT, or Breast Specific Gamma Imaging:**  
Used for staging or when cancer is suspected to have spread, not for routine measurement
  - **Blood Tests:**  
Not used for the measurements in your model, but sometimes used to check for tumor markers or overall health
- 

## How Are the Measurements Generated?

- **Image Analysis Software:**  
After imaging (mammogram, ultrasound, MRI), specialized computer software analyzes the digital images and calculates:
    - **Radius mean** (average distance from center to edge)
    - **Perimeter mean** (edge length)
    - **Area mean** (size)
    - **Texture mean** (variation in pixel intensity)
    - **Smoothness, compactness, concavity, symmetry, fractal dimension, etc.**
  - **Pathology Lab:**  
After a biopsy, the tissue is analyzed for cellular features (sometimes using advanced image analysis).
- 

## Summary Table: Tests and What They Provide

Test Type	What It Measures/Provides	Used For Model Features?
Mammogram	Lump size, shape, area, perimeter, texture	Yes
Ultrasound	Lump size, shape, edge characteristics	Yes
MRI	Detailed size, shape, complexity, internal features	Yes
Biopsy (Lab Analysis)	Cellular structure, edge smoothness, compactness	Yes
CT/PET/Other Scans	Staging, spread of cancer	No (not for basic features)
Blood Tests	Tumor markers, general health	No

---

## In Summary

- **Doctors perform imaging tests (mammogram, ultrasound, MRI) and/or a biopsy.**
- **Specialized software or lab analysis extracts the measurements (like radius, area, smoothness, etc.) from these tests.**
- **These values are then entered into your dashboard for prediction.**
- **Patients cannot obtain these values themselves; they come from clinical and lab tests performed by healthcare professionals.**

---

This is a **very common and important interview question**. Here's how you should answer if asked:

---

## Question:

“If doctors are doing lab tests and analyzing them to know whether a tumor is benign or malignant, what is the need for this AI/machine learning model?”

---

## Answer:

### 1. AI Complements, Not Replaces, Doctors

- The model is **not meant to replace doctors or lab tests**, but to **assist and enhance the diagnostic process**
- Even with lab tests and imaging, **human error, subjectivity, and diagnostic variability** can occur. Two experienced doctors might interpret the same image differently

---

### 2. Faster and More Consistent Diagnoses

- **AI models analyze data much faster** than traditional manual review, giving results in seconds instead of hours or days

- This is especially valuable in busy hospitals or places with a shortage of expert radiologists or pathologists.
- 

### 3. Reducing Errors and Variability

- **AI can reduce false positives and false negatives** by spotting subtle patterns in images or data that even skilled doctors might miss
  - This leads to **more accurate and consistent diagnoses** across different hospitals and regions
- 

### 4. A Second Opinion and Decision Support

- The model acts as a **“second reader” or digital assistant**. It can highlight suspicious cases for further review, or reassure doctors when the findings are clear
  - This is especially useful for **borderline or challenging cases**, where the diagnosis is not obvious.
- 

### 5. Early Detection and Personalized Care

- **AI can detect early-stage cancers** or subtle changes in tissue that may be invisible to the human eye
  - Early detection is critical for improving survival rates and outcomes
  - AI can also help **personalize treatment** by analyzing patterns and predicting molecular markers or responses to therapy
- 

### 6. Efficiency and Workload Reduction

- By automating routine analysis, AI **frees up doctors’ time** for more complex cases and patient care
  - This is especially helpful as the volume of medical imaging and data continues to grow.
- 

### 7. Objective, Data-Driven Insights



- AI provides **objective, reproducible results** based on large datasets, reducing the influence of fatigue, bias, or subjective judgment
- 

## Example to Use in Your Answer:

Even after a biopsy or imaging, the diagnosis can sometimes be unclear or subject to interpretation. AI models can quickly analyze all the measurements and images, providing a consistent, objective prediction. This helps doctors double-check their findings, reduce errors, and make faster, more confident decisions—especially in early or difficult cases.

---

## Summary Table: Human vs. AI in Breast Cancer Diagnosis

Aspect	Human Only (Traditional)	With AI Model (Your Project)
Speed	Hours/Days	Seconds/Minutes
Consistency	Varies by doctor	High (same for every case)
Error Risk	Human error possible	Reduced (AI + human = fewer errors)
Early Detection	Sometimes misses subtle signs	Can spot subtle/early changes
Second Opinion	Needs another doctor	AI provides instant second opinion
Workload	High, time-consuming	Reduced, more efficient

---

## References to Support Your Answer:

- AI improves accuracy, reduces errors, and speeds up diagnosis
  - AI acts as a second reader, supporting—not replacing—doctors
  - AI can detect subtle patterns and early-stage cancers that may be missed by the human eye
  - AI brings consistency and objectivity to the diagnostic process
- 

### In summary:

AI models are needed to make breast cancer diagnosis **faster, more accurate, more consistent, and less dependent on individual human judgment**. They support doctors, help catch cancers earlier, reduce unnecessary procedures, and improve patient outcomes. The best results come from **collaboration between AI and medical professionals**

The **Breast Cancer Wisconsin (Diagnostic) dataset** is calculated using advanced image analysis on samples collected from patients. Here's a detailed breakdown of **how the values are obtained and calculated**:

---

## How Is This Dataset Calculated?

### 1. Sample Collection: Fine Needle Aspirate (FNA)

- A **fine needle aspirate (FNA)** is performed on a breast mass. This is a minimally invasive procedure where a thin needle is used to extract cells from a suspicious breast lump for examination

### 2. Imaging and Digitization

- The extracted cells are placed on a microscope slide.
- The slide is **digitized** using a high-resolution camera attached to a microscope, creating a digital image of the cell nuclei

### 3. Feature Extraction Using Image Processing

- **Computer algorithms** analyze the digital image to measure various characteristics of the cell nuclei.
- **Key features calculated include**
  - **Radius:** Mean distance from the center to the edge of the nucleus.
  - **Texture:** Standard deviation of gray-scale pixel values (measures variation in intensity).
  - **Perimeter:** Length around the nucleus.
  - **Area:** Total number of pixels within the nucleus boundary.
  - **Smoothness:** Local variation in radius lengths (how smooth or rough the edge is).
  - **Compactness:** Calculated as  $(\text{perimeter}^2 / \text{area}) - 1.0$  -  $1.0(\text{perimeter}^2 / \text{area}) - 1.0$ , indicating how compact or spread out the nucleus is.

- **Concavity:** Severity of concave portions of the contour (how much the edge curves inward).
- **Concave points:** Number of distinct inward curves on the edge.
- **Symmetry:** How symmetric the nucleus shape is.
- **Fractal dimension:** Describes the complexity of the edge (using a “coastline approximation” method).

## 4. Aggregation of Measurements

- For each patient/sample, the computer calculates:
  - The **mean** value for each feature (e.g., `radius_mean`)
  - The **standard error** (`_se`) for each feature
  - The **worst (largest)** value (`_worst`) for each feature

## 5. Labeling

- Each sample is labeled by a pathologist as **malignant (M)** or **benign (B)** based on the microscopic appearance and clinical information.

---

## Summary Table: Example Features and Their Calculation

Feature	How It's Calculated (from image)
<code>radius_mean</code>	Average distance from center to edge of nucleus
<code>texture_mean</code>	Std. deviation of pixel gray values in nucleus
<code>perimeter_mean</code>	Total length around the nucleus
<code>area_mean</code>	Number of pixels inside the nucleus
<code>smoothness_mean</code>	Local variation in radius lengths
<code>compactness_mean</code>	$(\text{perimeter}^2 / \text{area}) - 1.0$
<code>concavity_mean</code>	Severity of concave portions of the edge
<code>concave points_mean</code>	Number of concave points on the edge
<code>symmetry_mean</code>	Degree of symmetry of the nucleus
<code>fractal_dimension_mean</code>	Complexity of the edge (fractal analysis)

---

## Key Points

- **All features are computed automatically** by image analysis software from digitized microscope images of cell samples
- **No manual measurement**—the process is objective and repeatable.
- The dataset contains **30 features per sample** (10 features  $\times$  3 types: mean, standard error, worst).
- **Diagnosis label** (benign or malignant) is assigned by a medical expert.

---

### In summary:

The values in the dataset are calculated by analyzing digital images of breast cell samples (from FNA), using computer vision algorithms to extract quantitative features that describe the shape, size, and texture of cell nuclei. These features are then used by machine learning models to predict whether a tumor is benign or malignant

Here is a **detailed, step-by-step explanation** of your R Shiny app code for breast cancer detection using SVM:

---

## 1. Loading Libraries

r

```
library(tidyverse)
```

```
library(caret)
```

```
library(shiny)
```

```
library(shinydashboard)
```

```
library(ggplot2)
```

```
library(DT)
```

```
library(pROC)
```

```
library(e1071)
```

**Purpose:**

These libraries provide tools for data manipulation (`tidyverse`), machine learning (`caret`, `e1071`), web app development (`shiny`, `shinydashboard`), visualization (`ggplot2`, `pROC`), and interactive tables (`DT`).

---

## 2. Data Loading and Preparation

r

```
data <- read.csv("../breast-cancer-wisconsin-data.csv")
data <- data %>% select(-id)

data <- data %>% select(diagnosis, radius_mean, texture_mean, perimeter_mean, area_mean,
                        smoothness_mean, compactness_mean, concavity_mean,
                        concave.points_mean, symmetry_mean, fractal_dimension_mean)

data$diagnosis <- factor(data$diagnosis, levels = c("B", "M"), labels = c("Benign", "Malignant"))

if (sum(is.na(data)) > 0) {
  stop("Data contains missing values.")
}
```

**Explanation:**

- Loads the CSV dataset.
  - Removes the `id` column (not needed for prediction).
  - Keeps only the relevant features and the target (`diagnosis`).
  - Converts `diagnosis` to a factor with labels "Benign" and "Malignant".
  - Checks for missing data and stops if any is found.
-

### 3. Splitting Data into Training and Test Sets

r

```
set.seed(123)

index <- createDataPartition(data$diagnosis, p = 0.7, list = FALSE)

train_data <- data[index, ]

test_data <- data[-index, ]
```

#### Explanation:

- Sets a random seed for reproducibility.
  - Splits the data: 70% for training, 30% for testing, maintaining the proportion of benign/malignant cases.
- 

### 4. Training the SVM Model

r

```
svm_model <- svm(diagnosis ~ ., data = train_data, kernel = "radial", cost = 1, gamma = 0.1, probability = TRUE)
```

#### Explanation:

- Trains a Support Vector Machine (SVM) using the training data.
  - Uses a radial basis function (RBF) kernel for non-linear separation.
  - `cost` and `gamma` are hyperparameters controlling model flexibility.
  - `probability = TRUE` enables probability predictions for ROC analysis.
- 

### 5. User Interface (UI) Design

r

```
ui <- dashboardPage(

  dashboardHeader(title = "Breast Cancer Detection", titleWidth = 250),
```

```

dashboardSidebar(
  width = 250,
  sidebarMenu(
menuItem("Dashboard", tabName = "dashboard", icon = icon("chart-line")),
menuItem("About", tabName = "about", icon = icon("info-circle")),
menuItem("Help", tabName = "help", icon = icon("question-circle"))
  ),
  dashboardBody(
tags$head(tags$style(HTML('...'))), # Custom pastel theme
    tabItems(
      # Dashboard tab
      tabItem(tabName = "dashboard",
        fluidRow(
          box(title = "Input Features", width = 6, solidHeader = TRUE,
            numericInput("radius_mean", ...),
            numericInput("texture_mean", ...),
            # ...other numericInputs for all 10 features
            actionButton("predict_btn", "Predict", ...),
            h4("Prediction Result:", ...),

```

```

        textOutput("prediction_result", inline = TRUE),

        textOutput("prediction_status")

    ),

    box(plotOutput("rocPlot"), width = 6),

    box(DTOutput("confMatrix"), width = 6)

)

),

# About tab

tabItem(tabName = "about", ...), # Info about breast cancer and the app

# Help tab

tabItem(tabName = "help", ...) # Instructions and usage help

)

)

)

```

## Explanation:

- **Sidebar:** Navigation between Dashboard, About, and Help.
- **Dashboard Tab:**
  - Input boxes for the 10 features (user enters measurements).
  - "Predict" button to trigger prediction.
  - Displays prediction result, ROC curve, and confusion matrix.
- **About Tab:** Educational info about breast cancer and the app.



- **Help Tab:** Instructions for using the app.

---

## 6. Server Logic

r

```
server <- function(input, output, session) {  
    # Prediction  
  
    output$prediction_result <- renderText({  
  
        req(input$predict_btn)  
  
        new_data <- data.frame(  
  
            radius_mean = input$radius_mean,  
  
            texture_mean = input$texture_mean,  
  
            perimeter_mean = input$perimeter_mean,  
  
            area_mean = input$area_mean,  
  
            smoothness_mean = input$smoothness_mean,  
  
            compactness_mean = input$compactness_mean,  
  
            concavity_mean = input$concavity_mean,  
  
            concave.points_mean = input$concave.points_mean,  
  
            symmetry_mean = input$symmetry_mean,  
  
            fractal_dimension_mean = input$fractal_dimension_mean  
  
        )  
  
        pred <- predict(svm_model, new_data)  
  
        paste("The predicted diagnosis is:", pred)
```

```
})
```

```
# ROC Curve
```

```
output$rocPlot <- renderPlot({
```

```
  prob <- attr(predict(svm_model, test_data, probability = TRUE), "probabilities")[, "Malignant"]
```

```
  roc_curve <- roc(test_data$diagnosis, prob, levels = c("Benign", "Malignant"))
```

```
  plot(roc_curve, main = "ROC Curve", col = "blue", lwd = 2)
```

```
})
```

```
# Confusion Matrix
```

```
output$confMatrix <- renderDT({
```

```
  predictions <- predict(svm_model, test_data)
```

```
  confusionMatrix(predictions, test_data$diagnosis)$table
```

```
})
```

```
}
```

## Explanation:

- **Prediction:**

- Waits for the "Predict" button.
- Collects user input into a data frame.
- Uses the trained SVM model to predict diagnosis.
- Displays the result ("Benign" or "Malignant").

- **ROC Curve:**
  - Calculates probability predictions for the test set.
  - Plots the ROC curve to show model performance.
- **Confusion Matrix:**
  - Predicts on the test set.
  - Displays a table showing true/false positives/negatives.

---

## 7. Running the App

```
r  
shinyApp(ui = ui, server = server)
```

### Explanation:

- Launches the Shiny app, making it interactive in a web browser.

---

## Summary Table

Step	What Happens
Load Libraries	Import all needed R packages
Load Data	Read CSV, select features, encode labels, check for missing values
Split Data	Divide into training and test sets
Train Model	Fit SVM model to training data
UI Design	Build dashboard with input fields, predict button, and output panels
Server Logic	Handle predictions, plot ROC, show confusion matrix
Run App	Start the web app

### In summary:

This app lets users enter tumor measurements, predicts if a tumor is benign or malignant using a trained SVM model, and displays model performance and educational information—all in a user-friendly dashboard.

Here’s a detailed comparison of the most common algorithms used for breast cancer prediction, based on your project context and recent scientific studies:

## Comparison of Algorithms for Breast Cancer Prediction

Algorithm	Description & Strengths	Typical Accuracy	Notes and Recent Findings
<b>Support Vector Machine (SVM)</b>	Finds the best boundary between classes, works well for high-dimensional data, robust to overfitting.	97–99%	Often achieves highest accuracy on the Wisconsin dataset. Fastest testing time <sup>15</sup> .
<b>Random Forest (RF)</b>	Ensemble of decision trees, handles non-linearity and feature interactions, robust to noise.	96–98%	Performs well, sometimes slightly less accurate than SVM or ANN <sup>15</sup> .
<b>Logistic Regression (LR)</b>	Simple linear classifier, interpretable, fast, works well for linearly separable data.	89–96%	Lower accuracy than SVM/RF/ANN, but easy to interpret <sup>15</sup> .
<b>Decision Tree (DT)</b>	Tree-based model, easy to interpret, can overfit on small datasets.	86–94%	Good for understanding feature importance, but less robust alone <sup>15</sup> .
<b>k-Nearest Neighbor (kNN)</b>	Classifies based on closest data points, simple, sensitive to feature scaling and irrelevant features.	92–97%	Fastest training time, but performance drops with noisy or high-dimensional data <sup>15</sup> .
<b>Artificial Neural Network (ANN)</b>	Mimics brain-like structure, good for complex patterns, needs more data and tuning.	98–99%	Achieved highest accuracy in some studies (up to 98.57%) <sup>1</sup> .
<b>Gradient Boosting (GB/GBM/XGBoost)</b>	Ensemble boosting method, builds strong model from weak learners, handles complex data well.	74–99%	Top performer in some studies, but can be slower to train <sup>135</sup> .
<b>Naive Bayes (NB)</b>	Probabilistic model, assumes feature independence, fast, works well with small data.	90–92%	Simpler, less accurate than SVM/ANN/RF <sup>5</sup> .
<b>Ensemble Methods</b>	Combines multiple models (e.g., SVM+RF+ANN), often boosts accuracy and robustness.	Up to 99%	Some studies show ensemble methods outperform individual models <sup>1</sup> .
<b>Tree Augmented Naive Bayes (TAN)</b>	Improved Naive Bayes with feature dependencies, better accuracy than plain NB.	~90–95%	TAN with boosting further improves accuracy <sup>2</sup> .

# Key Findings from Recent Studies

- **SVM:** Frequently outperforms other models on the Wisconsin dataset, achieving up to 97.2–99% accuracy<sup>15</sup>.
- **ANN/Deep Learning:** Can reach the highest accuracy (up to 98.57% or more) but requires more data and tuning<sup>1</sup>.
- **Random Forest & Gradient Boosting:** Both are strong performers, with Gradient Boosting sometimes leading, especially with feature engineering<sup>135</sup>.
- **Logistic Regression & Naive Bayes:** Simpler, more interpretable, but generally less accurate than ensemble or kernel-based methods<sup>5</sup>.
- **Ensemble Methods:** Combining models (e.g., SVM+RF+ANN) can further boost accuracy and reliability<sup>1</sup>.
- **Tree Augmented Naive Bayes (TAN):** Outperforms standard Naive Bayes when feature dependencies matter<sup>2</sup>.

---

## Summary Table: Algorithm Comparison

Algorithm	Accuracy (Typical)	Strengths	Weaknesses
SVM	97–99%	High accuracy, robust	Needs parameter tuning
ANN	98–99%	Best for complex data	Needs more data, less interpretable
Random Forest	96–98%	Handles non-linearity, robust	Can be slow with many trees
Gradient Boosting	74–99%	Top accuracy with tuning	Slower training, complex
Logistic Regression	89–96%	Simple, interpretable	Not for complex boundaries
Decision Tree	86–94%	Easy to interpret	Can overfit
kNN	92–97%	Simple, fast training	Sensitive to noise, scaling
Naive Bayes	90–92%	Fast, works with small data	Assumes feature independence
Ensemble Methods	Up to 99%	Combines strengths of models	More complex, harder to interpret
TAN (Bayesian)	90–95%	Handles dependencies	More complex than NB

---

## References

- [1](#) Nature Scientific Reports, 2024: SVM, ANN, RF, LR, DT, kNN, XGBoost, Ensemble[1](#)
- [2](#) PMC: TAN, BAN, Bayes Network[2](#)
- [3](#) WAOCP: Gradient Boosting, SVM, ANN, DT, LR, kNN, NB[35](#)
- [4](#) ScienceDirect: Comparative assessment of ML algorithms[46](#)

---

### In summary:

SVM, ANN, Random Forest, and Gradient Boosting are consistently among the top performers for breast cancer prediction. The best choice depends on your data size, need for interpretability, and computational resources. Ensemble methods and deep learning can further improve accuracy but may require more data and tuning.