

FINAL PROJECT REPORT

EUPHORIA GENX

TOPIC NAME : CANCER CELL PREDICTION (SVM)

SUBMITTED BY : Trinanjan Kumar Baul

Rajarshi Bhattacharya

Madhurima Saha

Ahana Biswas

Utsov Roy

Damayanti Roy

Sk. Aman Gani

SUBMITTED TO : Partha Koley

COURSE NAME : AI – ML (using Python)

COLLEGE NAME : FUTURE INSTITUTE OF
ENGINEERING AND MANAGEMENT

INDEX

1. Abstract	3
2. Acknowledgement	3-4
3. SDK	4-5
4. Model	5-7
5. Machine Learning	7-8
6. Machine Learning Life Cycle	8-11
7. Supervised and Unsupervised	12
8. Python	12
9. Workflow project	13-14
10. SVM	14-16
11. SVC	16
12. Kernel	16-17
13. Gamma	17
14. C	17-18
15. DecisionBoundaryDisplay	18
16. Prediction of Breast Cancer Cells	18-20
17. Source code and Output	20-26
18. Conclusion	26-27
19. Future scope	27
20. References	27

1. ABSTRACT

Breast cancer stands as a predominant cause of mortality among women globally, necessitating effective tools for its prediction and early diagnosis. The complexity of medical data analysis makes the prediction of breast cancer a challenging task. The integration of machine learning (ML) algorithms has emerged as a valuable solution in assisting doctors and pathologists in decision-making processes and distinguishing between malignant and benign tumors. This study explores the application of Support Vector Machines (SVMs), a powerful ML algorithm, in breast cancer prediction.

Research indicates that ML techniques play a pivotal role in decision-making processes related to breast cancer prediction. By leveraging data mining techniques, the study proposes a method to reduce the reliance on conventional tests, such as MRI, mammogram, ultrasound, and biopsy, by focusing on detecting the presence of the risk of breast cancer. The proposed method utilizes a dataset available in the sklearn library, consisting of unique ID numbers, corresponding diagnoses (malignant/benign), and real-value features (parameters).

The SVM learning algorithm is employed to construct a predictive model capable of identifying whether a tumor is malignant (1) or benign (0). The proposed method not only aids in efficient risk assessment but also contributes to the optimization of diagnostic processes by minimizing the number of tests required. The study emphasizes the significance of machine learning, particularly SVM, as a valuable tool in breast cancer prediction, paving the way for more streamlined and accurate diagnostic procedures. Keywords include Machine Learning, SVM, Kernel, Python, and Artificial Intelligence, reflecting the technological aspects essential to the proposed methodology.

2. ACKNOWLEDGEMENT

SDT(Software Development Tools):-

Software development tools are computer programs used by software development teams to create, debug, manage and support applications, frameworks, systems, and other programs. These tools are also commonly referred to as software programming tools. In some cases, one tool can house multiple functions.

Numpy :-

Numpy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python. Besides its obvious scientific uses, Numpy can also be used as an efficient multi-dimensional container of generic data.

Pandas :-

It is built on the top of the NumPy library which means that a lot of structures of NumPy are used or replicated in Pandas. The data produced by Pandas are often used as input for plotting functions of Matplotlib, statistical analysis in SciPy, and machine learning algorithms in Scikit-learn.

Matplotlib :-

Matplotlib is an amazing visualization library in Python for 2D plots of arrays. Matplotlib is a multi-platform data visualization library built on NumPy arrays and designed to work with the broader SciPy stack. It was introduced by John Hunter in the year 2002.

Pyplot :-

Pyplot is a sub-module of the Matplotlib module with an interface like that of Matplotlib. It has various graph-generating features that use Python and take the maximum advantage of open-source and being free. Each pyplot function makes some change to a figure: e.g., creates a figure, creates a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels etc.

Seaborn :-

Seaborn aims to make visualization the central part of exploring and understanding data. Seaborn is a library for making statistical graphics in Python. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data

3. SOFTWARE DEVELOPMENT KIT

An SDK (software development kit) is a collection of tools for developing applications for specific hardware/software or in a certain programming language. With some interpreted languages, the SDK can be identical to the run-time environment. SDKs typically include an integrated development environment (IDE), which serves as the central programming interface.

The "google.colab" library provides functionality for tasks such as 4 importing and exporting files, installing Python packages, managing Colab sessions, and connecting to external services like Google Drive and Google Sheets. Some of the common tasks that can be performed using the "google.colab"

LIBRARY USED:-

Importing and exporting files:-The library allows you to upload and download files to and from the Colab environment. For example, you can use the "files.upload()" function to upload files from your local machine to Colab, and the "files.download()" function to download files from Colab to your local machine.

Installing Python packages:-The library provides a way to install Python packages directly from within the Colab environment using the "!pip install" command.

Managing Colab sessions:- The library allows you to manage the lifecycle of a Colab session. You can use functions like "drive.mount()" to mount your Google Drive, "drive.flush_and_unmount()" to flush and unmount the Google Drive, and "os.kill()" to terminate the current session.

Connecting to external services: The library provides functionality to connect to external services like Google Drive and Google Sheets, allowing you to read and write data to these services from within a Colab notebook.

Interacting with Colab UI:- The library allows you to interact with the Colab user interface programmatically, for example, by using the "IPython.display" module to display images, videos, and other media in the output of a Colab cell. Overall, while Google Colab does not have a standalone SDK, the "google.colab" library provides a convenient way to interact with the Colab environment programmatically and automate various tasks within Colab notebooks. You can import the "google.colab" library in your Python code and use its functions to perform operations within the Colab environment.

4. MODEL

Waterfall Model

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are –

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

Spiral Model

The biggest problem we face in the waterfall model is that taking a long duration to complete the product, and the software became outdated. To solve this problem, we have a new approach, which is known as the Spiral model. The spiral model is also known as the cyclic model.

In this model, we create the application module by module and handed over to the customer so that they can start using the application at a very early stage. And we prepare this model only when the module is dependent on each other. In this model, we develop the application in the stages because sometimes the client gives the requirements in between the process.

The different phases of the spiral model are as follows:

- **Requirement analysis**
- **Design**
- **Coding**
- **Testing and risk analysis**

Incremental model :-

The incremental model is not a separate model. It is necessarily a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software. The SDLC process is repeated, with each release adding more functionality until all requirements are met. In this method, each cycle acts as the maintenance phase for the previous software release. Modification to the incremental model allows development cycles to overlap. After that subsequent cycle may begin before the previous cycle is complete.

RAD model :-
The Rapid Application Development Model was first proposed by IBM in the 1980s. The RAD model is a type of incremental process model in which there is an extremely short development cycle. When the requirements are fully understood and the component-based construction approach is adopted then the RAD model is used. Various phases in RAD are Requirements Gathering, Analysis and Planning, Design, Build or Construction, and finally Deployment.

The critical feature of this model is the use of powerful development tools and techniques. A software project can be implemented using this model if the project can be broken down into small modules wherein each module can be assigned independently to separate teams. These modules can finally be combined to form the final product. Development of each module involves the various basic steps as in the waterfall model i.e. analyzing, designing, coding, and then testing, etc. as shown in the figure. Another striking feature of this model is a short time span i.e. the time frame for delivery (time-box) is generally 60-90 days.

V model :-

The V-model is a type of SDLC model where process executes in a sequential manner in V-shape. It is also known as Verification and Validation model. It is based on the association of a testing phase for each corresponding development stage.

Development of each step is directly associated with the testing phase. The next phase starts only after completion of the previous phase i.e. for each development activity, there is a testing activity corresponding to it.

The V-Model is a software development life cycle (SDLC) model that provides a systematic and visual representation of the software development process. It is based on the idea of a "V" shape, with the two legs of the "V" representing the progression of the software development process from requirements gathering and analysis to design, implementation, testing, and maintenance.

Prototyping Model has different phases, which are as follows:

- **Requirement analysis**
- **feasibility study**
- **Create a prototype**
- **Prototype testing**
- **Customer review and approval**
- **Design**
- **Coding**
- **Testing**
- **Installation and maintenance**

5. MACHINE LEARNING

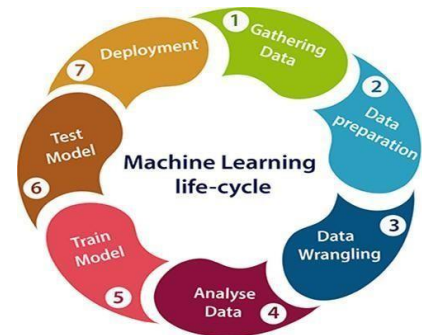
Machine learning is a branch of artificial intelligence that employs a variety of statistical, probabilistic and optimization techniques that allows computers to “learn” from past examples and to detect hard-to-discern patterns from large, noisy or complex data sets. This capability is particularly well-suited to medical applications, especially those that depend on complex proteomic and genomic measurements. As a result, machine learning is frequently used in cancer diagnosis and detection. More recently machine learning has been applied to cancer prognosis and prediction. This latter approach is particularly interesting as it is part of a growing trend towards personalized, predictive medicine. In assembling this review, we conducted a broad survey of the different types of machine learning methods being used, the types of data being integrated and the performance of these methods in cancer prediction and prognosis. A number of trends are noted, including a growing dependence on protein biomarkers and microarray data, a strong bias towards applications in prostate and breast cancer, and a heavy reliance on “older” technologies such artificial neural networks (ANNs) instead of more recently developed or more easily interpretable machine learning methods. A number of published studies also appear to lack an appropriate level of validation or testing. Among the better designed and

validated studies, it is clear that machine learning methods can be used to substantially (15–25%) improve the accuracy of predicting cancer susceptibility, recurrence, and mortality. At a more fundamental level, it is also evident that machine learning is also helping to improve our basic understanding of cancer development and progression.

6. MACHINE LEARNING LIFE CYCLE

The 6 steps in a standard machine learning life cycle:

1. Planning
2. Data Preparation
3. Model Engineering
4. Model Evaluation
5. Model Deployment
6. Monitoring and Maintenance



Each phase in the machine learning cycle follows a **quality assurance** framework for constant improvement and maintenance by strictly following requirements and constraints. Learn more about quality assurance by reading the **CRISP-ML(Q)** blog.

For non-technical individuals and managers, check out our short course on **Understanding Machine Learning** fundamentals. It will help them understand machine learning in general, modeling, and deep learning (AI).

1. Planning

The planning phase involves assessing the scope, success metric, and feasibility of the ML application. You need to understand the business and how to use machine learning to improve the current process. For example: do we require machine learning? Can we achieve similar requests with simple programming?

You also need to understand the cost-benefit analysis and how you will ship the solution in multiple phases. Furthermore, you need to define clear and measurable success metrics for business, machine learning models (Accuracy, F1 score, AUC), and economic (key performance indicators).

Finally, you need to create a feasibility report.

It will consist of the information about:

- **Availability of the data:** do we have enough data available to train the model? Can we get a constant supply of new and updated data? Can we use synthetic data to reduce the cost?
- **Applicability:** will this solution solve the problem or improve the current process? Can we even use machine learning to solve this issue?
- **Legal constraints:** do we have permission from the local government to implement this solution? Are we following an ethical way of collecting the data? What will be the impact of this application on society?
- **Robustness and scalability:** is this application robust enough? Is it scalable?
- **Explainability:** can we explain how the machine learning model is coming up with the results? Can we explain the deep neural networks' inner workings?
- **Availability of resources:** do we have enough computing, storage, network, and human resources? Do we have qualified professionals?

2. Data Preparation

The data preparation section is further divided into four parts: data procurement and labeling, cleaning, management, and processing.

• Data collection and labeling

We need first to decide how we will collect the data by gathering the internal data, open-source, buying it from the vendors, or generating synthetic data. Each method has pros and cons, and in some cases, we get the data from all four methodologies.

After collection, we need to label the data. Buying cleaned and labeled data is not feasible for all companies, and you may also need to make changes to the data selection during the development process. That is why you cannot buy it in bulk and why the data can eventually be useless for the solution.

The data collection and labeling require most of the company resources: money, time, professionals, subject matter experts, and legal agreements.

• Data Cleaning

Next, we will clean the data by imputing missing values, analyzing wrong-labeled data, removing outliers, and reducing the noise. You will create a data pipeline to automate this process and perform data quality verification.

• Data processing

The data processing stage involves feature selection, dealing with imbalanced classes, feature engineering, data augmentation, and normalizing and scaling the data.

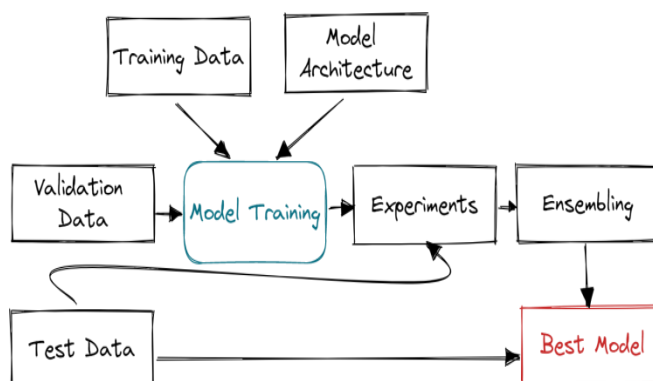
For reproducibility, we will store and version the metadata, data modeling, transformation pipelines, and feature stores.

- **Data management**

Finally, we will figure out data storage solutions, data versioning for reproducibility, storing metadata, and creating ETL pipelines. This part will ensure a constant data stream for model training.

3. Model Engineering

In this phase, we will be using all the information from the planning phase to build and train a machine learning model. For example: tracking model metrics, ensuring scalability and robustness, and optimizing storage and compute resources.



1. Build effective model architecture by doing extensive research.
2. Defining model metrics.
3. Training and validating the model on the training and validation dataset.
4. Tracking experiments, metadata, features, code changes, and machine learning pipelines.
5. Performing model compression and ensembling.
6. Interpreting the results by incorporating domain knowledge experts.

We will be focusing on model architecture, code quality, machine learning experiments, model training, and ensembling.

The features, hyper parameters, ML experiments, model architecture, development environment, and metadata are stored and versioned for reproducibility.

Learn about the steps involved in model engineering by taking the **Machine Learning Scientist with Python** career track. It will help you master the necessary skills to land a job as a machine learning engineer.

4. Model Evaluation :

Now that we have finalized the version of the model, it is time to test various metrics. Why? So that we can ensure that our model is ready for production.

We will first test our model on a test dataset and make sure we involve subject matter experts to identify the error in the predictions.

We also need to ensure that we follow industrial, ethical, and legal frameworks for building AI solutions.

Furthermore, we will test our model for robustness on random and real-world data. Making sure that the model inferences fast enough to bring the value.

Finally, we will compare the results with the planned success metrics and decide on whether to deploy the model or not. In this phase, every process is recorded and versioned to maintain quality and reproducibility.

5. Model Deployment



In this phase, we deploy machine learning models to the current system. For example: introducing automatic warehouse labeling using the shape of the product. We will be deploying a computer vision model into the current system, which will use the images from the camera to print the labels.

Generally, the models can be deployed on the cloud and local server, web browser, package as software, and edge device. After that, you can use API, web app, plugins, or dashboard to access the predictions. In the deployment process, we define the inference hardware. We need to make sure we have enough RAM, storage, and computing power to produce fast results. After that, we will evaluate the model performance in production using A/B testing, ensuring user acceptability.

The deployment strategy is important. You need to make sure that the changes are seamless and that they have improved the user experience. Moreover, a project manager should prepare a disaster management plan. It should include a fallback strategy, constant monitoring, anomaly detection, and minimizing losses.

6. Monitoring and Maintenance

After deploying the model to production we need to constantly monitor and improve the system. We will be monitoring model metrics, hardware and software performance, and customer satisfaction.

The monitoring is done completely automatically, and the professionals are notified about the anomalies, reduced model and system performance, and bad customer reviews.

After we get a reduced performance alert, we will assess the issues and try to train the model on new data or make changes to model architectures. It is a continuous process.

7. SUPERVISED AND UNSUPERVISED LEARNING

ML encompasses a broad range of tasks and methods. Supervised learning tasks have a known available outcome to predict, such as presence of a tumour, length of survival, or treatment response. Unsupervised learning identifies patterns and subgroups within data where there is no clear outcome to predict. It is often used for more exploratory analysis.

Supervised learning, as the name indicates, has the presence of a supervisor as a teacher. Basically, supervised learning is when we teach or train the machine using data that is well-labelled. Which means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples (data) so that the supervised learning algorithm analyses the training data (set of training examples) and produces a correct outcome from labelled data.

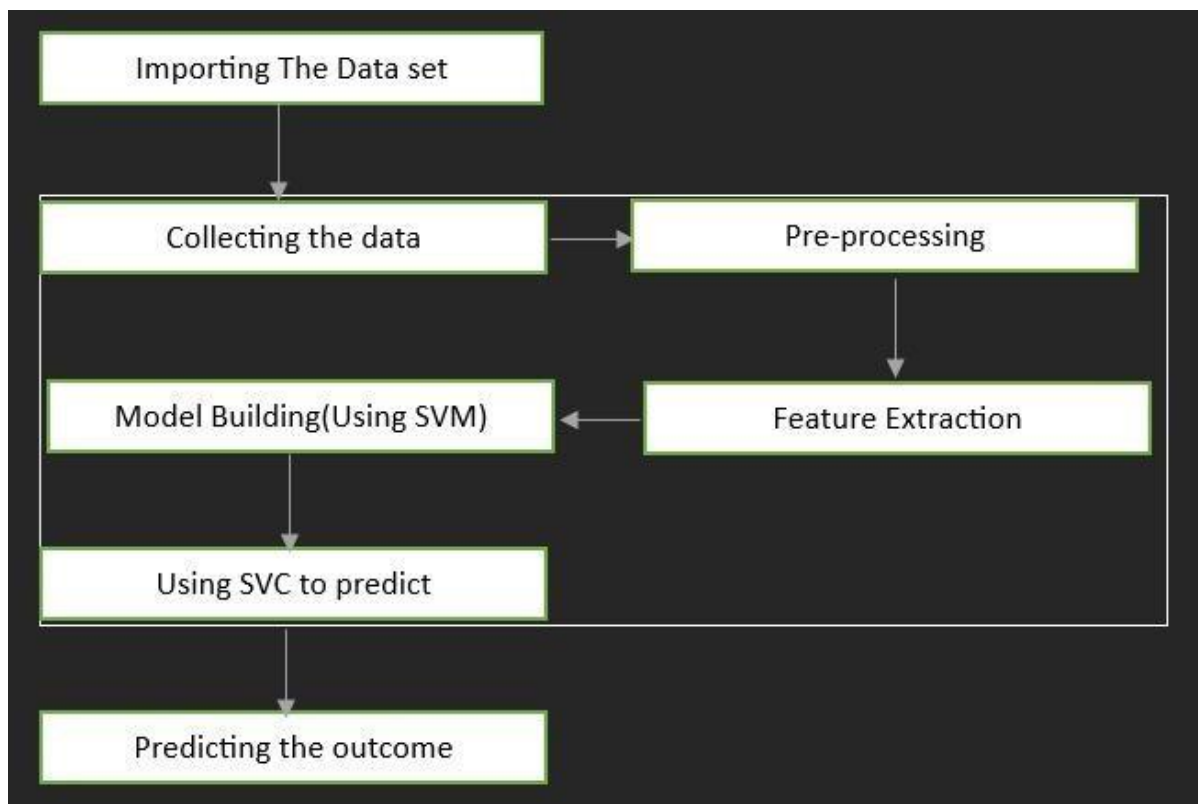
Unsupervised learning is the training of a machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns, and differences without any prior training of data.

8. PYTHON

Python is the best choice for building machine learning models due to its ease of use, extensive framework library, flexibility and more.

Python brings an exceptional amount of power and versatility to machine learning environments. The language's simple syntax simplifies data validation and streamlines the scraping, processing, refining, cleaning, arranging and analyzing processes, thereby making collaboration with other programmers less of an obstacle. Python also offers a vast ecosystem of libraries that take much of the monotonous routine function writing tasks out of the equation to free developers up to focus on code and reduces the chances for error when programming.

9. WORKFLOW PROJECT



Workflow Management Breast Cancer Prediction

Workflow Management in breast cancer prediction includes cancer diagnosis and detection. More recently machine learning has been applied to cancer prognosis and prediction. A machine learning (ML) algorithm helps a lot to take decisions and to perform diagnosis from the data collected by medical field. Various research show that ML techniques are helpful for decision making in breast cancer prediction. Here's a general outline of a typical outflow of management system of breast cancer prediction in medical fields.

Data Collection :- Data related to various factors that influence the tumour, such as radius, texture, concavity, perimeter, fractal dimension, compactness error, concave points error, symmetry error, etc are collected from the patients and integrated into the workflow management system. This data can be collected through testing and diagnosing multiple patients with tumours.

Preprocessing :- The collected data is pre-processed to clean and transform it into a format suitable for analysis. This may involve data cleaning, normalization, aggregation, and feature extraction to reduce noise and ensure data quality.

Data Analysis :- The pre-processed data is analyzed using various statistical and machine learning techniques to identify patterns, trends, and correlations between different variables. For example, machine learning algorithms such as decision trees, random forests, and neural networks can be used to predict tumour type based on patient data and new diagnosis.

Cancer Cell Prediction :- Based on the analysis results, the workflow management system can generate cancer cell prediction models that can forecast the treatment of different cancer patients. These models can be continuously updated with new data to improve their accuracy overtime.

Monitoring and Feedback :- The workflow management system can continuously monitor the actual tumours and yield data and compare it with the predicted results. This feed back loop allows for ongoing validation and refinement of the prediction models, and helps patients make informed decisions.

Reporting and Visualization :- The workflow management system can generate reports and visualizations to provide doctors, patients and other medical staffs with a clear understanding of the cell prediction results, tumour types, and performance metrics. This can help doctors evaluate the effectiveness of their diagnostic strategies and make data-driven decisions for future.

Continuous Improvement :- The workflow management system can be continuously improved by incorporating new data sources, updating prediction models, and refining decision support algorithms based on feedback from doctors and other medical professionals. This iterative process helps ensure that the system remains accurate, reliable, and relevant over time.

Overall, an effective workflow management system for cancer cell prediction in agricultural systems involves the integration of data collection, preprocessing, analysis, prediction, monitoring, reporting, and continuous improvement components to enable efficient and data-driven crop management practices.

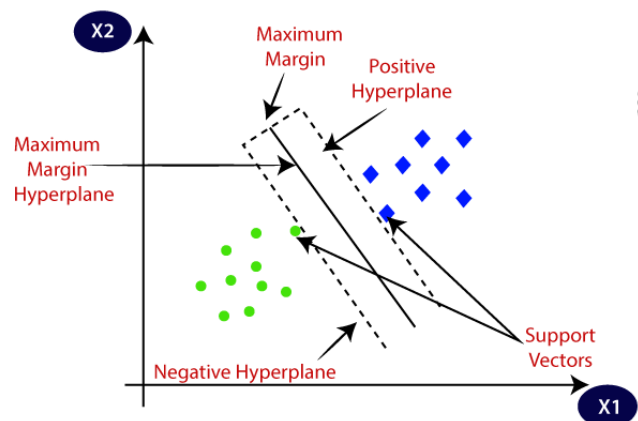
10. SVM

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks.

SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships.

Support Vector Machine (SVM) is a supervised machine learning algorithm used for both classification and regression. SVM can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. The

main objective of the SVM algorithm is to find the optimal hyperplane in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.



❖ Implementing SVM in Python

SVM Kernels

In practice, SVM algorithm is implemented with kernel that transforms an input data space into the required form. SVM uses a technique called the kernel trick in which kernel takes a low dimensional input space and transforms it into a higher dimensional space. In simple words, kernel converts non-separable problems into separable problems by adding more dimensions to it. It makes SVM more powerful, flexible and accurate. The following are some of the types of kernels used by SVM.

- **Linear Kernel**

It can be used as a dot product between any two observations. The formula of linear kernel is as below –

$$K(x, x_i) = \sum (x * x_i)$$

- **Polynomial Kernel**

It is more generalized form of linear kernel and distinguish curved or nonlinear input space. Following is the formula for polynomial kernel –

$$k(X, X_i) = 1 + \sum (X * X_i)^d$$

- **Radial Basis Function (RBF) Kernel**

RBF kernel, mostly used in SVM classification, maps input space in indefinite dimensional space. Following formula explains it mathematically –

$$K(x, x_i) = \exp(-\gamma \sum (x - x_i)^2)$$

❖ **Types of SVM :**

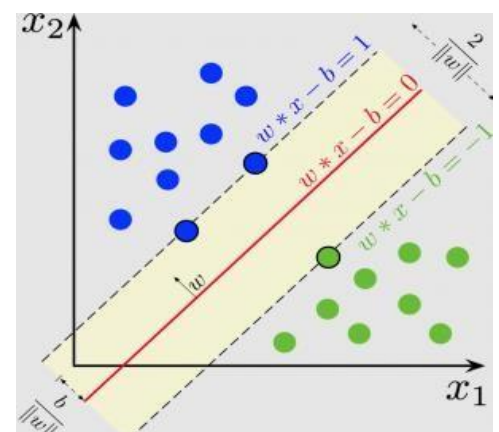
SVM can be of two types:

- I. **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
- II. **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

11. SVC

An SVM classifier, or support vector machine classifier, is a type of machine learning algorithm that can be used to analyze and classify data. A support vector machine is a supervised machine learning algorithm that can be used for both classification and regression tasks. The Support vector machine classifier works by finding the hyperplane that maximizes the margin between the two classes. The Support vector machine algorithm is

also known as a max-margin classifier. Support vector machine is a powerful tool for machine learning and has been widely used in many tasks such as hand-written digit recognition, facial expression recognition, and text classification.



12. KERNEL

SVM algorithms use a set of mathematical functions that are defined as the kernel. The function of kernel is to take data as input and transform it into the required form. Different SVM algorithms use different types of kernel functions. These functions can be different types. For example **linear, nonlinear, polynomial, radial basis function (RBF), and sigmoid**. The most used type of kernel function is **RBF**. Because it has localized and finite

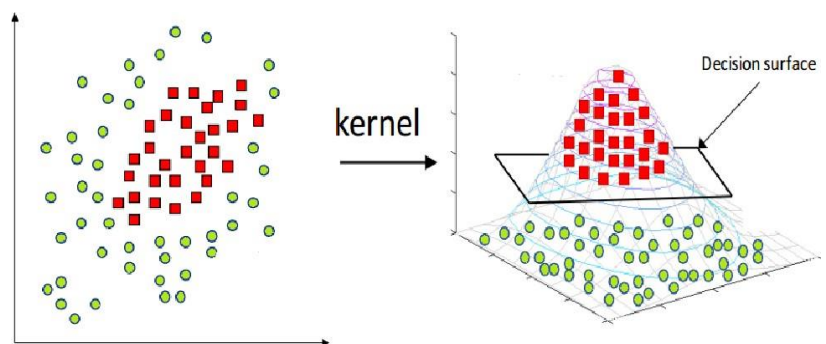
response along the entire x-axis. The kernel functions return the inner product between two points in a suitable feature space. Thus by defining a notion of similarity, with little computational cost even in very high-dimensional spaces.

$$K(\bar{x}) = \begin{cases} 1 & \text{if } \|\bar{x}\| \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

Kernel Function is a method used to take data as input and transform it into the required form of processing data. “Kernel” is used due to a set of mathematical functions used in Support Vector Machine providing the window to manipulate the data. So, Kernel Function generally transforms the training set of data so that a non-linear decision surface is able to transform to a linear equation in a higher number of dimension spaces. Basically, It returns the inner product between two points in a standard feature dimension.

13. GAMMA

- Gamma is used when we use the Gaussian RBF kernel.
- Gamma is a hyper parameter which we have to set before training

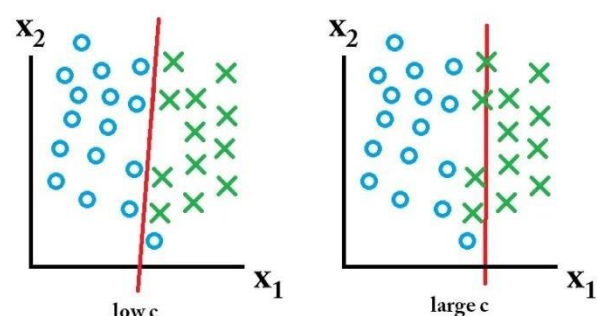


model. Gamma decides that how much curvature we want in a decision boundary.

- Gamma high means more curvature.
- Gamma low means less curvature.

14. C

C- It is a hypermeter in SVM to control error. What does that mean to control error or margin? Let's understand with visualization.



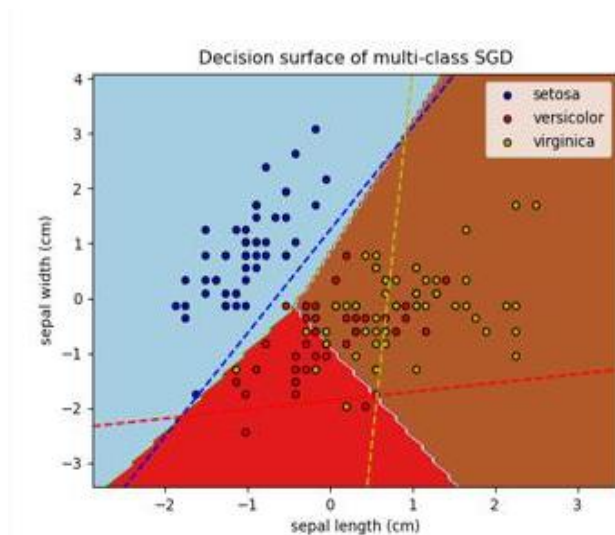
You can see if we have low C means low error and if we have large C means large error.

In low C we have only one error but in case of large C, we have four errors.
let's see some other dataset.

C is a **hyperparameter** which is set before the training model and used to control error and Gamma is also a hyperparameter which is set before the training model and used to give curvature weight of the decision boundary.

15. DECISIONBOUNDARYDISPLAY

While training a classifier on a dataset, using a specific classification algorithm, it is required to define a set of hyper-planes, called Decision Boundary, that separates the data points into specific classes, where the algorithm switches from one class to another. On one side a decision boundary, a data points is more likely to be called as class A — on the other side of the boundary, it's more likely to be called as class B.



The decision boundary, comes up as nonlinear and non-smooth. The function works with any Scikit-learn estimator, even a neural network. Here is the decision boundary with the MLP Classifier estimator of Scikit-learn, which models a densely-connected neural network (with user-configurable parameters).

Input data that should be only 2-dimensional. Number of grid points to use for plotting decision boundary. Higher values will make the plot look nicer but be slower to render. Extends the minimum and maximum values of X for evaluating the response function. Plotting method to call when plotting the response.

16. PREDICTION OF BREAST CANCER CELL

Breast cancer is one of the most lethal and heterogeneous disease in this present era that causes the death of enormous number of women all over the world. It is the second largest disease that is responsible of women death . There are various

machine learning and data mining algorithms that are being used for the prediction of breast cancer.

Machine learning is using data to answer questions. So Prediction, or inference, is the step where we get to answer some questions. This is the point of all this work, where the value of machine learning is real.

- **Logistic Regression**

It is a supervised learning algorithm that includes more dependent variables. The response of this algorithm is in the binary form. Logistics regression can provide the continuous outcome of a specific data. This algorithm consists of statistical model with binary variables

- **K-Nearest Neighbor (KNN)**

This algorithm is used in pattern recognition. It is a good approach for breast cancer prediction. In order to recognize the pattern, each class has given an equal importance. K Nearest Neighbor extract the similar featured data from a large dataset. On the basis of features similarity we classify a big dataset.

- **Decision Tree (DT)**

Decision tree [37] is based on classification and regression model. Dataset is divided into smaller number of subsets. These smaller set of data can make prediction with the highest level of precision. Decision tree method includes CART, C4.5, C5.0 and conditional tree.

- **Support Vector Machine (SVM)**

It is a supervised learning algorithm which is used for both classification and regression problems. It consists of theoretical and numeric functions to solve the regression problem. It provides the highest accuracy rate while doing prediction of large dataset. It is a strong machine learning technique that is based on 3D and 2D modelling

- **Random Forest (RF)**

Random Forest algorithm is based on supervised learning that is used to solve both classification and regression problems. It is a building block of machine learning that is used for prediction of new data on the basis of previous dataset

- **K Mean Algorithm**

K mean is clustering algorithm that provides the partition of data in the form of small clusters. Algorithm is used to find out the similarity between different data points.

Data points exactly consist of at least one cluster that is most suitable for the evaluation of big dataset .

- **C Mean Algorithm**

Clusters are identified on the similarity basis. Cluster that consist of similar data point belongs to one single family. In C mean algorithm each data point belongs to one single cluster. It is mostly used in medical images segmentation and disease prediction.

17. SOURCE CODE & OUTPUT

Importing the libraries and the dataset:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.datasets import load_breast_cancer
from sklearn.inspection import DecisionBoundaryDisplay
from sklearn.svm import SVC
```

Printing the imported data set:

```
cancer=load_breast_cancer()
print(cancer)
```

Output:-

```
{'data': array([[1.799e+01, 1.038e+01, 1.228e+02, ..., 2.654e-01, 4.601e-01,
                1.189e-01],
                [2.057e+01, 1.777e+01, 1.329e+02, ..., 1.860e-01, 2.750e-01,
                8.902e-02],
                [1.969e+01, 2.125e+01, 1.300e+02, ..., 2.430e-01, 3.613e-01,
                8.758e-02],
                ...,
                [1.660e+01, 2.808e+01, 1.083e+02, ..., 1.418e-01, 2.218e-01,
                7.820e-02], [2.060e+01, 2.933e+01, 1.401e+02, ..., 2.650e-01,
                4.087e-01,1.240e-01],
```

```
[7.760e+00, 2.454e+01, 4.792e+01, ..., 0.000e+00, 2.871e-01,  
 7.039e-02]], 'target': array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1,  
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0,  
 0, 0, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 1, 1, 1, 1, 0, 1, 0, 0,  
 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0,  
 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 1, 1, 0, 1, 1, 0, 1,  
 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0,  
 0, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,  
 1, 1, 0, 1, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 1,  
 1, 0, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,  
 0, 0, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0,  
 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1,  
 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
 0, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1,  
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1,  
 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0,  
 0, 1, 1, 1, 1, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0,  
 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0,  
 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1,  
 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 1, 1, 0,  
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 0, 1, 1, 1, 1,  
 1, 0, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0,  
 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1,  
 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1,  
 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1,  
 1, 1, 1, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
```

```
[  
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1]), 'frame': None, 'target_names':
```

```

array(['malignant',      'benign'],      dtype='<U9'),      'DESCR':      '..
_breast_cancer_dataset:\n\nBreast cancer wisconsin (diagnostic) dataset\n-----
-----\n\n**Data Set Characteristics:**\n\n      :Number of
Instances: 569\n\n :Number of Attributes: 30 numeric, predictive attributes and the
class\n\n :Attribute Information:\n - radius (mean of distances from center to points
on the perimeter)\n      - texture (standard deviation of gray-scale values)\n
- perimeter\n - area\n - smoothness (local variation in radius lengths)\n - compactness
(perimeter^2 / area - 1.0)\n      - concavity (severity of concave portions of the
contour)\n - concave points (number of concave portions of the contour)\n
      - symmetry\n      - fractal dimension ("coastline approximation" -
1)\n\n      The mean, standard error, and "worst" or largest (mean of the three\n
worst/largest values) of these features were computed for each image,\n resulting
in 30 features. For instance, field 0 is Mean Radius, field\n      10 is Radius SE, field 20
is Worst Radius.\n\n      - class:\n      - WDBC-Malignant\n      -
WDBC-Benign\n\n      :Summary      Statistics:\n\n
=====
Min      Max\n      =====
radius (mean):      6.981 28.11\n texture (mean):      9.71
39.28\n perimeter (mean):      43.79 188.5\n area (mean):
143.5 2501.0\n smoothness (mean):      0.053 0.163\n compactness
(mean):      0.019 0.345\n concavity (mean):      0.0 0.427\n
concave points (mean):      0.0 0.201\n symmetry (mean):      0.106
0.304\n fractal dimension (mean):      0.05 0.097\n radius (standard error):
0.112 2.873\n texture (standard error):      0.36 4.885\n perimeter (standard
error):      0.757 21.98\n area (standard error):      6.802 542.2\n
smoothness (standard error):      0.002 0.031\n compactness (standard error):
0.002 0.135\n concavity (standard error):      0.0 0.396\n concave points
(standard error):      0.0 0.053\n symmetry (standard error):      0.008 0.079\n
fractal dimension (standard error):      0.001 0.03\n radius (worst):      7.93
36.04\n texture (worst):      12.02 49.54\n perimeter (worst):
50.41 251.2\n area (worst):      185.2 4254.0\n smoothness (worst):
0.071 0.223\n compactness (worst):      0.027 1.058\n concavity (worst):
0.0 1.252\n concave points (worst):      0.0 0.291\n symmetry (worst):
0.156 0.664\n fractal dimension (worst):      0.055 0.208\n
=====
:Missing
Attribute Values: None\n\n :Class Distribution: 212 - Malignant, 357 - Benign\n\n
:Creator: Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian\n\n :Donor: Nick
Street\n\n :Date: November, 1995\n\nThis is a copy of UCI ML Breast Cancer
Wisconsin (Diagnostic) datasets.\nhttps://goo.gl/U2Uwz2\n\nFeatures are computed

```

from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree Construction Via Linear Programming." Proceedings of the 4th Midwest Artificial Intelligence and Cognitive Science Society, pp. 97-101, 1992], a classification method which uses linear programming to construct a decision tree. Relevant features were selected using an exhaustive search in the space of 1-4 features and 1-3 separating planes.

The actual linear program used to obtain the separating plane in the 3-dimensional space is that described in:

K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp.cs.wisc.edu/cdm/math-prog/cpo-dataset/machine-learn/WDBC/.. topic:: References

- W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on Electronic Imaging: Science and Technology, volume 1905, pages 861-870, San Jose, CA, 1993.

- O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and prognosis via linear programming. Operations Research, 43(4), pages 570-577, July-August 1995.

- W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994) 163-171.

'feature_names': array(['mean radius', 'mean texture', 'mean perimeter', 'mean area',

'mean smoothness', 'mean compactness', 'mean concavity',

'mean concave points', 'mean symmetry', 'mean fractal dimension',

'radius error', 'texture error', 'perimeter error', 'area error',

'smoothness error', 'compactness error', 'concavity error',

'concave points error', 'symmetry error',

'fractal dimension error', 'worst radius', 'worst texture',

'worst perimeter', 'worst area', 'worst smoothness',

'worst compactness', 'worst concavity', 'worst concave points',

'worst symmetry', 'worst fractal dimension'], dtype='<U23'), 'filename': 'breast_cancer.csv', 'data_module': 'sklearn.datasets.data'}

Creating the x and y axis:

```
x=cancer.data[:, : 2]
```

```
y=cancer.target
```



```
print(x)
```

```
print(y)
```

Output:-

```
[[17.99 10.38]
```

```
[20.57 17.77]
```

```
[19.69 21.25]
```

```
...
```

```
[16.6 28.08]
```

```
[20.6 29.33]
```

```
[ 7.76 24.54]]
```

```
[000000000000000000000000011100000000000000000
10000000001011111001001111010011110100
1010011100100011101100111001111011011
1111110001001110010100100110110111101
1111111101111001011001100111101100010
1011101100100001000101011010000110011
1011111001101100101111011111010000000
0000000111111010110110100111111111111
1011010111111111111110111010111100011
1101010111011111111000111111111100100
0100111110111110111011001111110111111
101111101101111111111111101001011111011
0101101011111111100111111011111111101
1111110101101111100101011111011010100
1110111111111110100111111111111111111
11111110000001]
```

Printing the Shape:

```
print(x.shape)
```

```
print(y.shape)
```

Output:-

```
(569, 2)
```

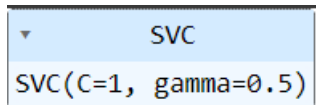
```
(569,)
```

Importing SVC Model(Using Kernel,gamma& C):

```
svm_model=SVC(kernel="rbf",gamma=.5,C=1)
```

```
svm_model.fit(x,y)
```

Output:-

A screenshot of a Jupyter Notebook cell showing the output of the SVC model. It displays a dropdown menu with 'SVC' selected, and below it, the text 'SVC(C=1, gamma=0.5)'.

ScatterPlot:

```
import matplotlib.colors
```

```
mycol=matplotlib.colors.ListedColormap(["red","green"])
```

```
DecisionBoundaryDisplay.from_estimator(svm_model,
```

```
x,
```

```
    response_method="predict",
```

```
    cmap=plt.cm.Spectral,
```

```
    xlabel=cancer.feature_names[0],
```

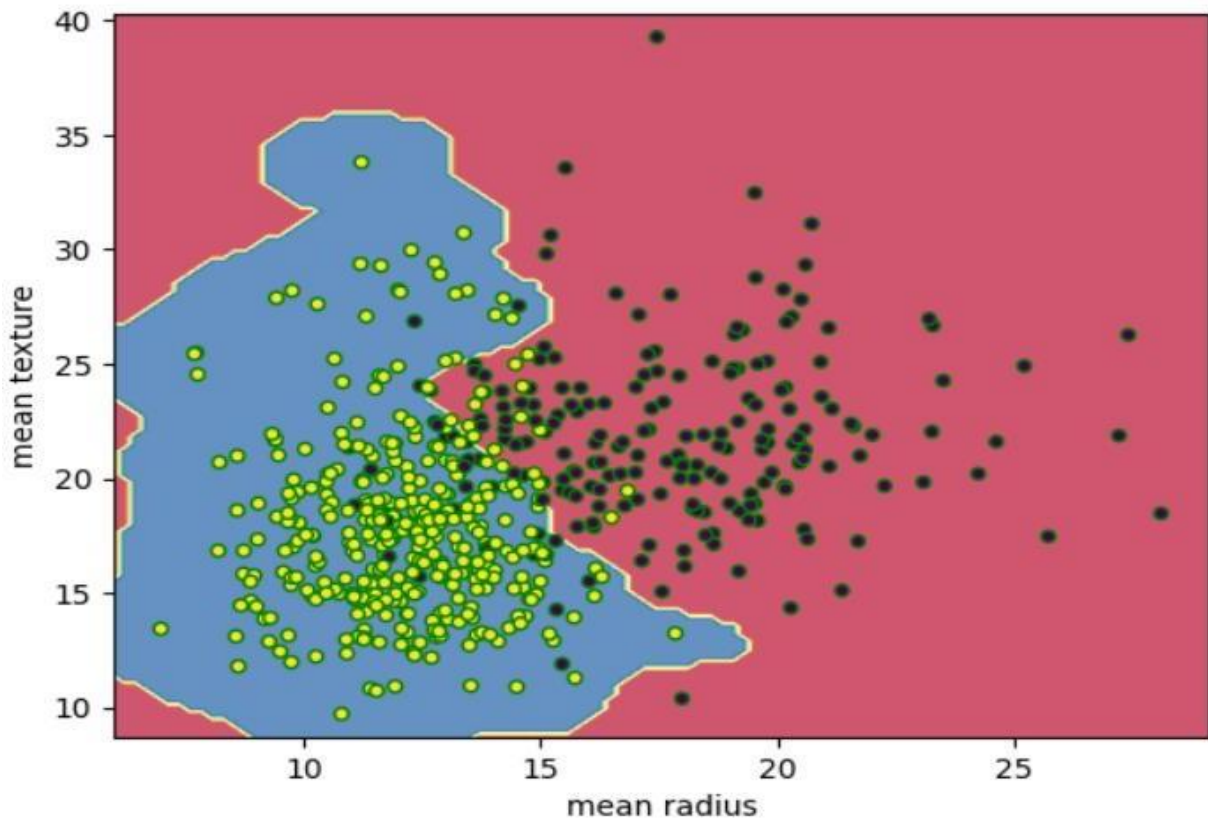
```
    ylabel=cancer.feature_names[1],
```

```
)
```

```
plt.scatter(x[:,0],x[:,1],c=y,s=20,edgecolors="green")
```

Output:-

<matplotlib.collections.PathCollection at 0x7e252ad2ada0>



Prediction :-

```
inp=[[10,15]]
yp=svm_model.predict(inp)
print(y_name[yp[0]])
```

Output :-

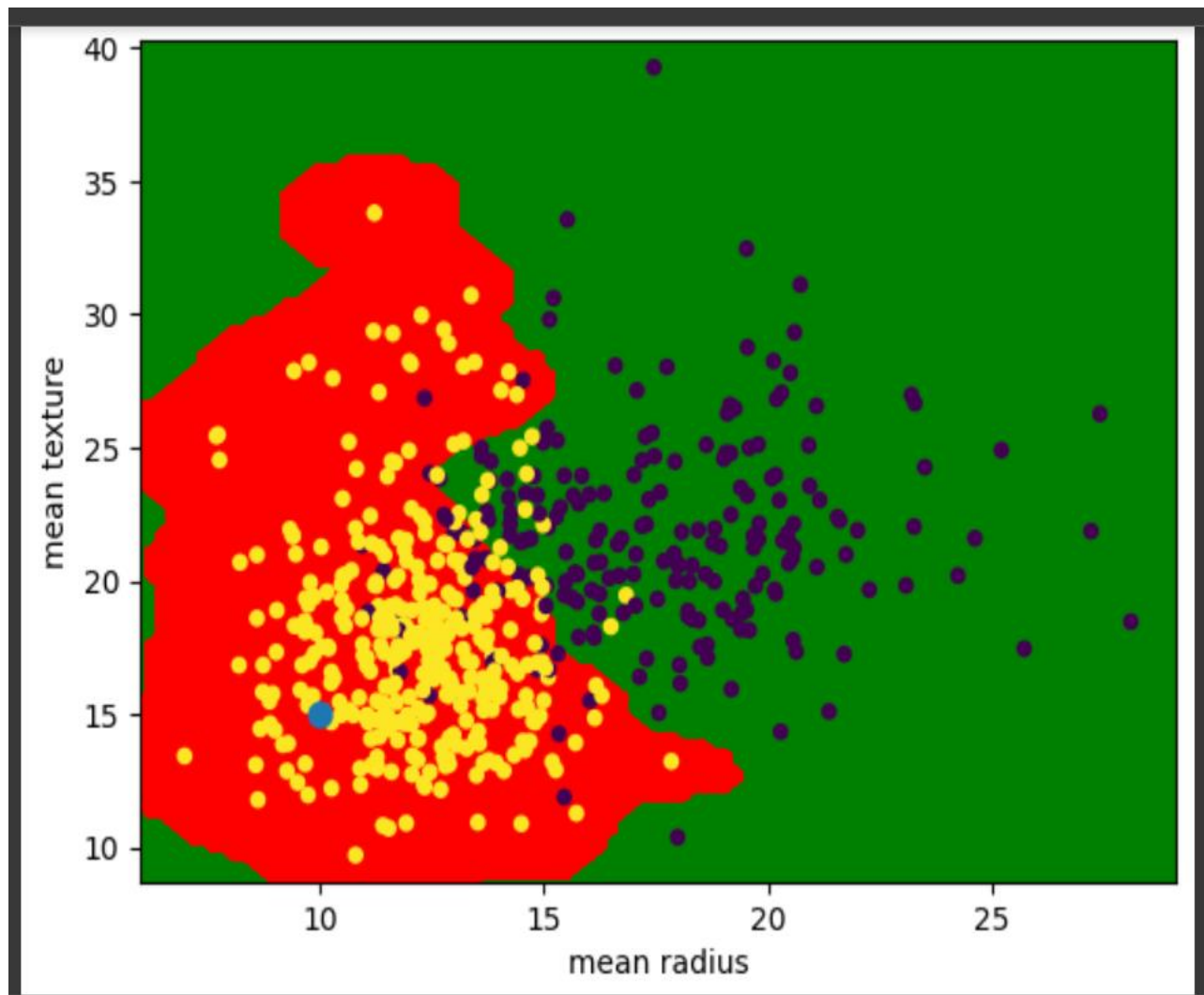
benign

Prediction through ScatterPlot :-

```
DecisionBoundaryDisplay.from_estimator(
    svm_model,
    x,
    response_method="predict",
    xlabel=cancer.feature_names[0],
    ylabel=cancer.feature_names[1],
    cmap=mycol
```

```
)  
  
radius=x[:,0]  
txt=x[:,1]  
plt.scatter(radius,txt,c=y,s=20)  
plt.scatter(10,15,s=60)
```

Output :-



18. CONCLUSION

Breast cancer is the important field of research and technology helps to reduce mortality rate caused by breast cancer. There are many ML algorithms introduced till now for analysis of medical datasets. It is essential for a medical diagnosis that the data on breast cancer be classified in a way that is both accurate and effective. Even though many numbers of methods have been developed to classify breast cancer data, there are still many obstacles to overcome, including accuracy. In order to address this issue, we put forth a model for the classification of data relating to breast cancer. In this paper we applied SVM, ML Classification technique. The proposed machine-learning approaches could predict breast cancer as the early detection of this disease could help slow down the progress of the disease and reduce the mortality rate through appropriate therapeutic interventions at the right time. Applying different machine learning approaches, accessibility to bigger datasets from different institutions (multi-centre study) and considering key features from a variety of relevant data sources could improve the performance of modelling.

In conclusion, the implementation of SVM can produce almost enough accuracy to be termed as a medically acceptable level of diagnostic accuracy for the dataset used. However, the dataset is not highly normalized resulted in an over-fitting problem due to the number of prominent outliers as seen while tuning the parameters. This limitation partly affected the accuracy of the model.

19. FUTURE SCOPE

AI is set to change the medical industry in the coming decades — it wouldn't make sense for pathology to not be disrupted too. Currently, ML models are still in the testing and experimentation phase for cancer prognoses. As datasets are getting larger and of higher quality, researchers are building increasingly accurate models.

Here's what a future cancer biopsy might look like:

You perform clinical tests, either at a clinic or at home. Data is inputted into a pathological ML system. A few minutes later, you receive an email with a detailed report that has an accurate prediction about the development of your cancer. While you might not see AI doing the job of a pathologist today, you can expect ML to replace your local pathologist in the coming decades, and it's pretty exciting! ML models still have a long way to go, most models still lack sufficient data and suffer from bias. Yet, something we are certain of is that **ML is the next step of pathology, and it will disrupt the industry.**

20. REFERENCES

- Builtin.com
- www.ncbi.nlm.nih.gov
- ascopubs.org
- geeksforgeeks.org
- www.researchgate.net
- www.sciencedirect.com
- journals.sagepub.com
- www.simplilearn.com
- www.javatpoint.com
- www.baeldung.com
- tutorialspoint.com
- ieeexplore.ieee.org
- vitalflux.com
- medium.com

Signature of the HOD

Signature of the Industrial
Project's Mentor