Graphical user interface

Description automatically generated with low confidence

Tumor Cancer Prediction

Report

T.A: Dr/ Alaa Tarek

Team members:

|  |  |
| --- | --- |
| Member: | ID |
| Heba Mohamed Saleh | 20201700961 |
| Yasmine Khaled atta | 20201701154 |
| Hesham Ayman Mohamed Ahmed | 20201700969 |
| Merna Ashraf Fayez | 20201701209 |
| Mohamed Ashraf Alywa | 20201700668 |
| Ziad Mohamed Esmaeil |  |

Content

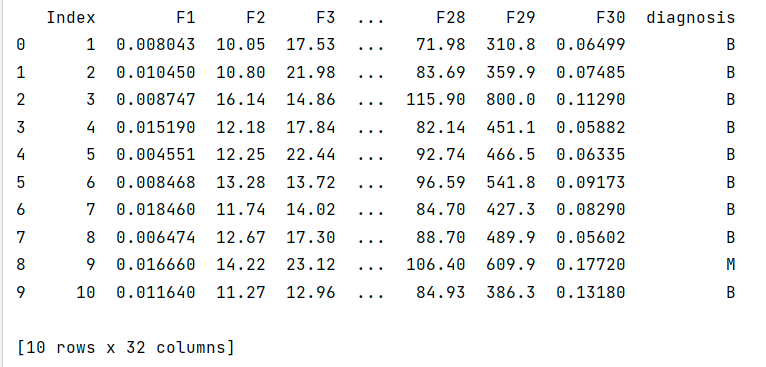
1. Introduction
2. Data Cleansing
3. Feature Selection
4. Splitting the dataset
5. Model Classifier
6. Voting classifier
7. save and load machine learning models
8. GUI for input file using tkinter

Our Dataset Definition:

For our analysis, we got a dataset contains 455 row each row consist of 30 independent

Features (F1 -> F30) and 1 dependent feature (diagnosis), which will be explained in the next section.

Dataset Sample:



Introduction

Cancer is one of the most lethal diseases in the present era that causes the death of enormous number of people. Cancer cells are categorized into benign and malignant cells. The benign cells do not spread to other parts, while malignant cells metastasize and are considered more destructive.

Using machine learning algorithms: Logistic Regression, Decision tree, SVM with linear kernel, SVM with polynomial kernel, Random Forest; we can help to predict that it's malignant or benignant.

Data Cleansing

Data cleansing is a very important process, it represent 60% form data scientists work. Without good data, there is no good model. So, we did the following preprocessing:

* First, removing the useless column (‘index’).



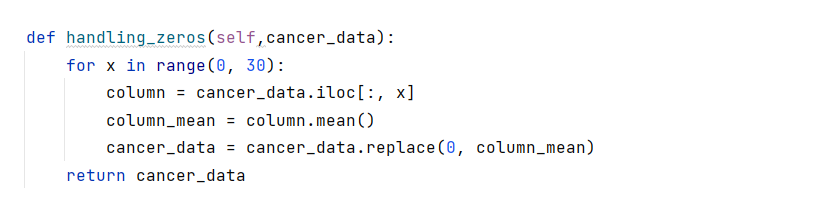
* Removing the Nan values in the dataset by the function (dropna).



* Removing the duplicated values by the function (drop duplicates).

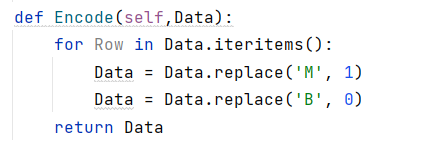


* Handling zeros by replacing each zero with the mean of its column.

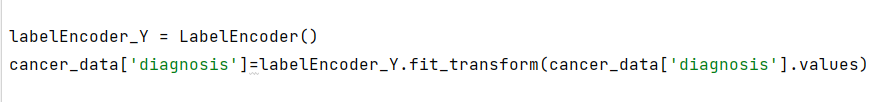


* Data conversion, we encoded the category data into numerical representation in target column (‘diagnosis’) from B or M to 0 or 1. We did it in 2 ways :

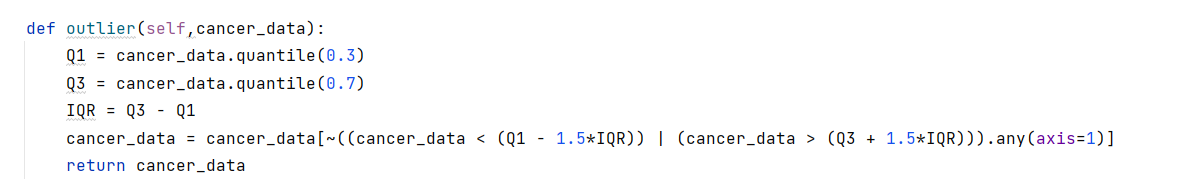
First:



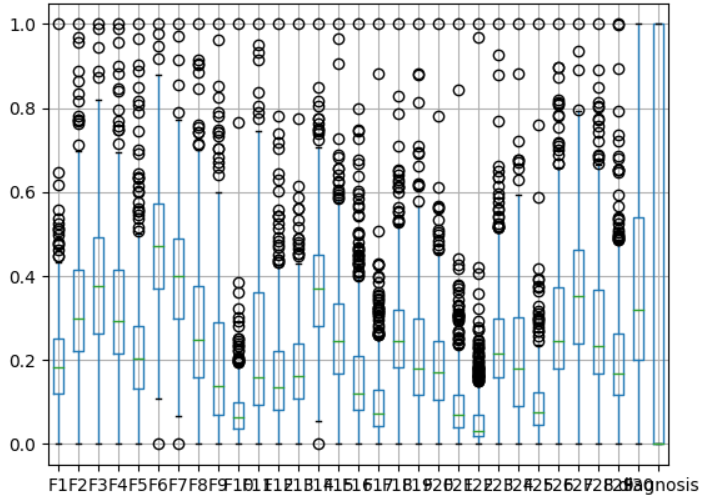
Second:



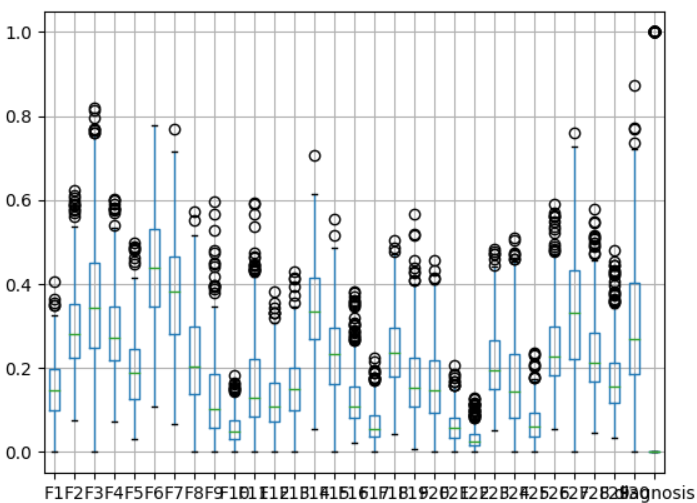
* Then, we are dealing with outliers by removing the rows which has outlier values, so the shape of the dataset becomes (307, 30).



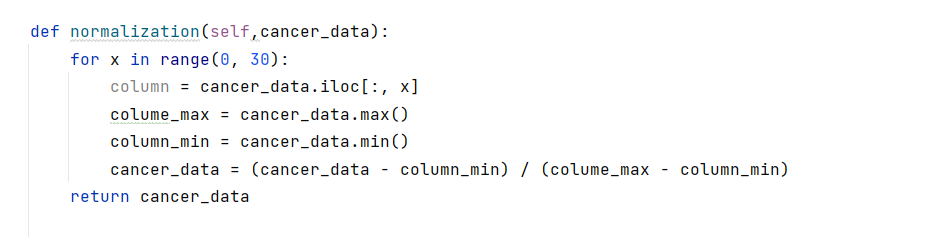
Box plot before removing the outliers:



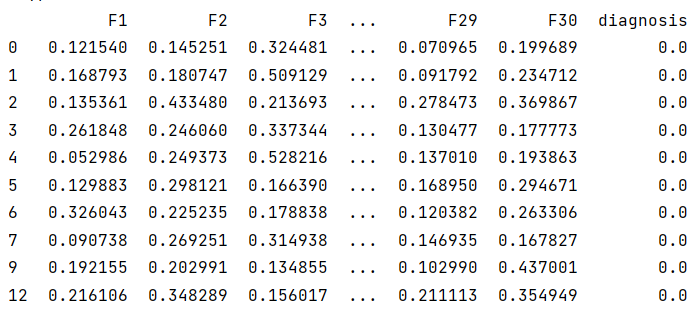
Box plot after removing the outliers:



* Data normalization, mapping each value in the dataset to a value in range [0, 1].



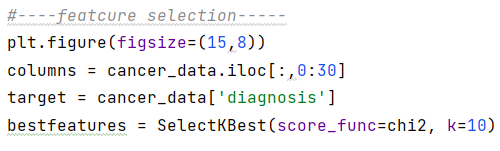
Dataset sample after the preprocessing:



Feature Selection

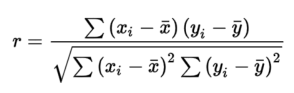
Feature selection is necessary because it: Reduces the time complexity, Avoid dimension explosion, improve model generalization and Avoid overfitting.

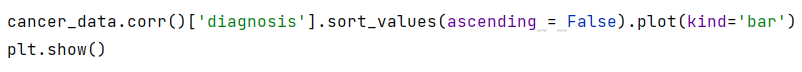
We used the built in function (SelectKBest) to choose the best features in the dataset.

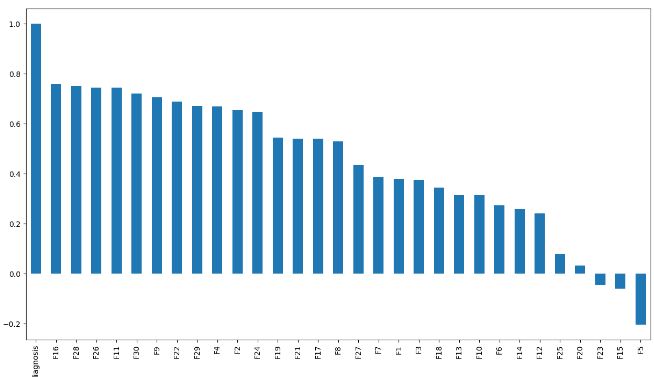


Correlation coefficients are indicators of the strength of the linear relationship between two different variables, x and y. A linear correlation coefficient that is greater than zero indicates a positive relationship. A value that is less than zero signifies a negative relationship. Finally, a value of zero indicates no relationship between the two variables x and y.

The Pearson’s correlation formula is:



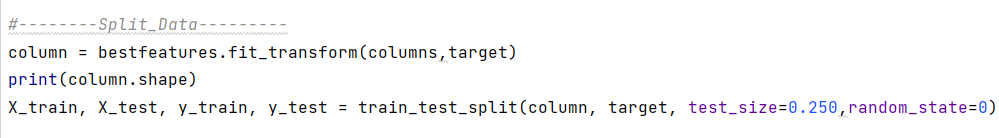




Splitting the dataset

In machine learning, it is a common practice to split your data into training set and testing set. As the name suggests, the training set is used for training the model and the testing set is used for testing the accuracy of the model

In our dataset the train set contains 341 rows and test set contains 114 rows.

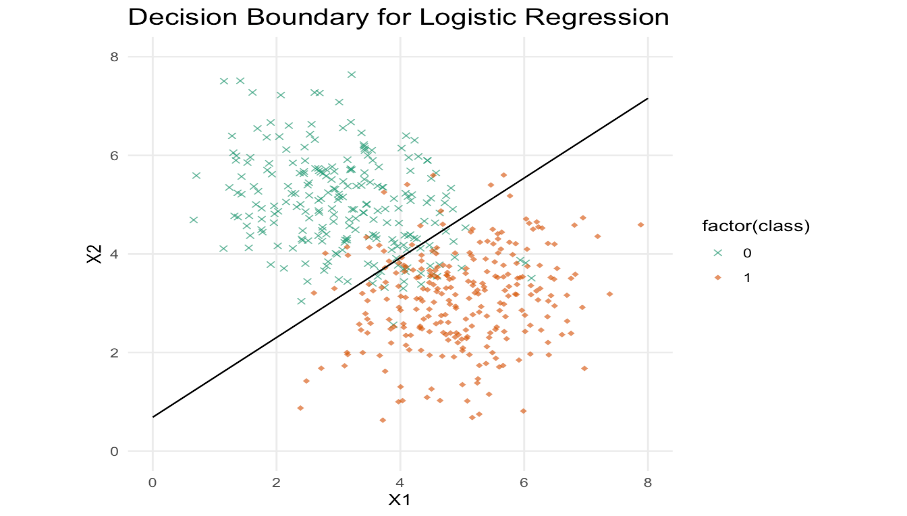


Model Classifiers

**We implement 5 models each one in a distinct class ..**

1-Logistic Regression

What is Logistic Regression? The logistic regression statistic modeling technique is used when we have a binary outcome variable. For example: given the parameters, will the student pass or fail? Will it rain or not? So, though we may have continuous or categorical independent variables, we can use the logistic regression modeling technique to predict the outcome when the outcome variable is binary.

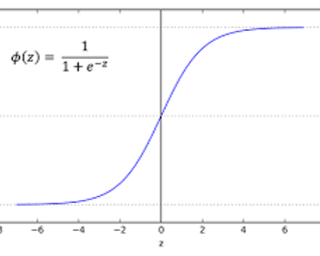


The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. The formula of the hypothesis i.e.

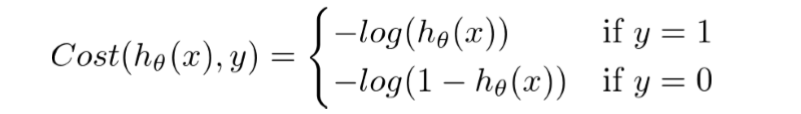
hΘ(x) = β₀ + β₁X

Math Prerequisites:

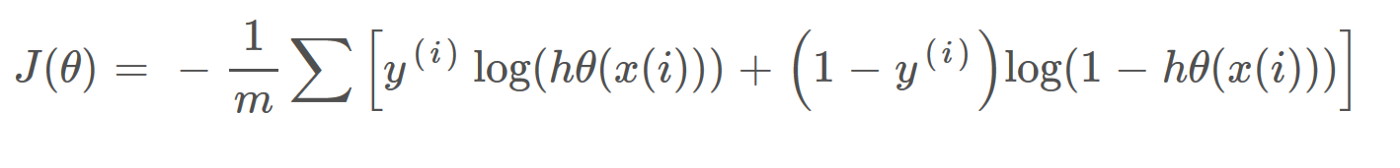
Sigmoid Function: The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the sigmoid function or the logistic function.



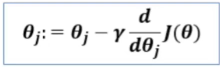
Cost Function: We learnt about the cost function J (θ) in the Linear regression, the cost function represents optimization objective i.e. we create a cost function and minimize it so that we can develop an accurate model with minimum error.

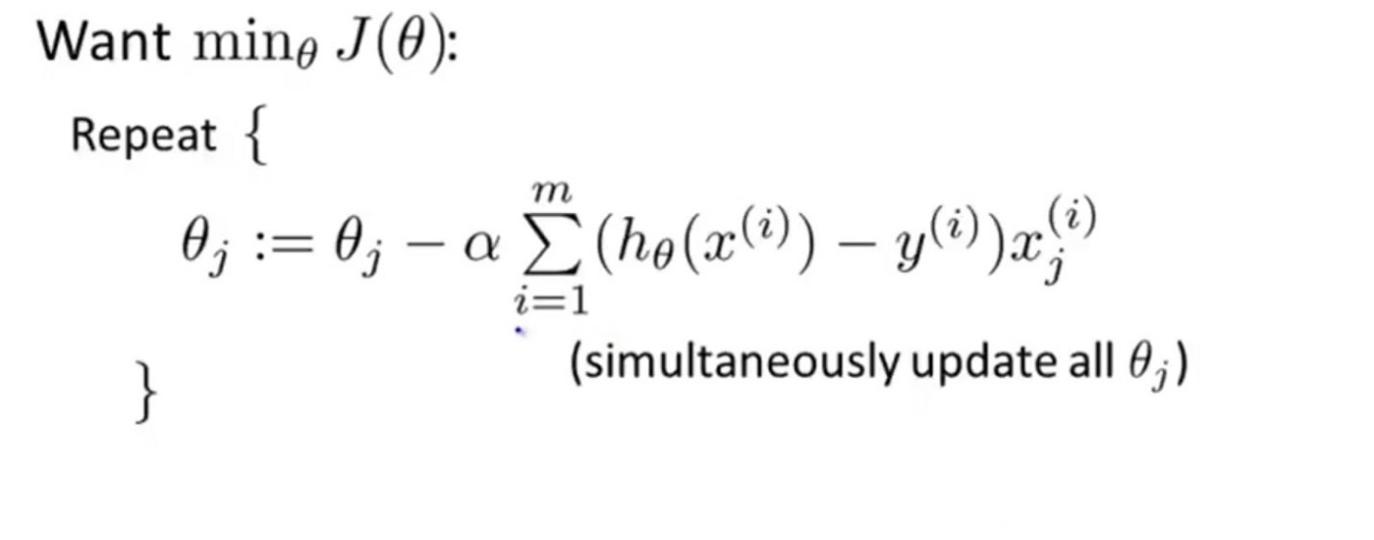


The above two functions can be compressed into a single function i.e.



Now to minimize our cost function we need to run the gradient descent function on each parameter i.e.





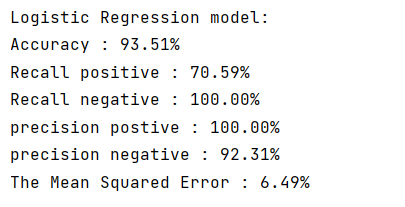


The above statement creates an instance of LogisticRegression and binds its references to the variable logistic\_regression.



.fit () takes x train, y train. Then it ﬁts the model and returns the model instance itself.

Logistic Regression Evaluation :



2-Decision Tree

* Decision Tree is a **Supervised learning technique**that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where**internal nodes represent the features of a dataset, branches represent the decision rules** and **each leaf node represents the outcome.**
* In a Decision tree, there are two nodes, which are the **Decision Node** and**Leaf Node.** Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.

Steps for Making decision tree:

1. Feature selection: Select a feature from the features of the training data as the split standard of the current node. (Different standards generate different decision tree algorithms.)
2. Decision tree generation: Generate internal node upside down based on the selected features and stop until the dataset can no longer be split.
3. Pruning: The decision tree may easily become overfitting unless necessary pruning (including pre-pruning and post-pruning) is performed to reduce the tree size and optimize its node structure.

Below diagram illustrate the basic flow of decision tree:

Diagram

Description automatically generated

Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as **Attribute selection measure or ASM.**By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

* **Information Gain )ID3)**
* **Gini Index**

We use information gain (ID3):

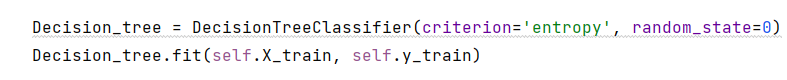
Information gain or IG is a statistical property that measures how well a given attribute separates the training examples according to their target classification. Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy

Entropy is a measure of the randomness in the information being processed.

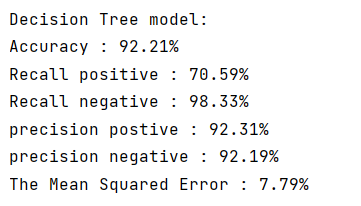
Graphical user interface, text, application

Description automatically generated

Code Implementation:



Decision Tree Evaluation :



3-Random Forest

* Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset.

Below diagram illustrate the basic flow of Random Forest:

***Diagram

Description automatically generated***

Steps involved in random forest algorithm:

Step 1: In Random forest n number of random records are taken from the

data set having k number of records.

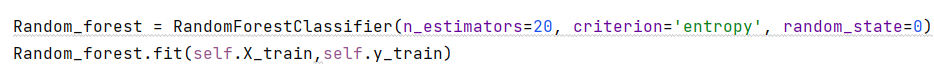
Step 2: Individual decision trees are constructed for each sample.

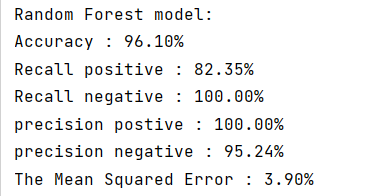
Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for

Classification and regression respectively.

Code Implementation:

Random forset Evaluation :



4-Support Vector Machine

* Straight lines are used to divide data into different classes. Actually, we can use multiple straight lines to divide data. The core idea of the SVM is to find a straight line and keep the point close to the straight line as far as possible from the straight line.
* the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a hyperplane. SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors. The distance between the vectors and the hyperplane is called as margin. And the goal of SVM is to maximize this margin. The hyperplane with maximum margin is called the optimal hyperplane.

Below diagram illustrate the basic flow of decision tree:

Diagram, schematic

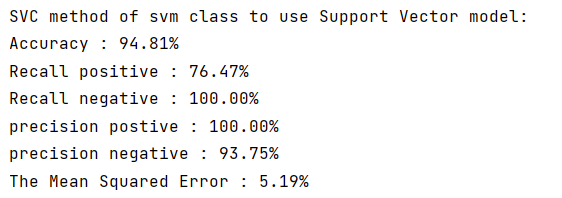
Description automatically generated

Code Implementation:

A picture containing text

Description automatically generated

SVM Evaluation:

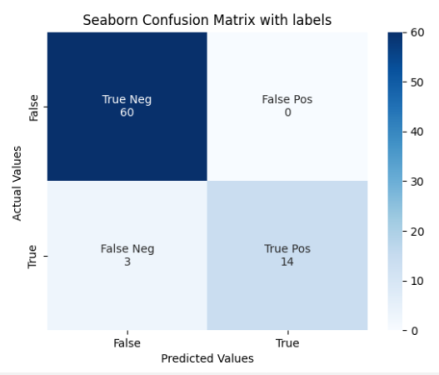


Models Evaluation

1. Confusion Matrix

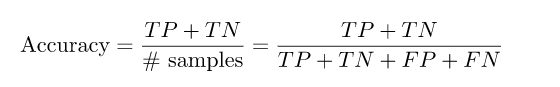
A confusion matrix is a n x n matrix (where n is the number of labels) used to describe the performance of a classification model. Each row in the confusion matrix represents an actual class whereas each column represents a predicted class.

Code Implementation:



1. Accuracy

The most simple and straightforward classification metric is accuracy. Accuracy measures the fraction of correctly classified observations. The formula is:



1. Precision

Precision is defined as the ratio of True Positives count to total True Positive count made by the model.

Precision = TP/ (TP+FP)

1. Recall

Recall is defined as the ratio of True Positives count to the total Actual Positive count.

Recall = TP/ (TP+FN)

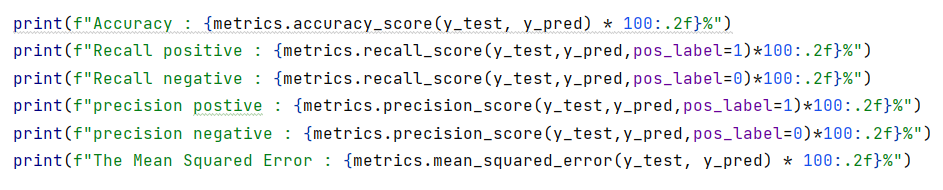
Recall is also called “True Positive Rate” or “sensitivity”.

1. Mean squared error

The mean squared error (MSE) tells you how close a regression line is to a set of points.

MSE formula = (1/n) \* Σ (actual – forecast) 2

Code Implementation:



Voting classifier

* A voting classifier is a machine learning estimator that trains various base models or estimators and predicts on the basis of aggregating the findings of each base estimator. The aggregating criteria can be combined decision of voting for each estimator output.
* After classifying the inputted sample by the three independently trained classifiers, the voting module combines their outputs to assign the inputted sample to the most frequent output (class).

Diagram

Description automatically generated

* Code Implementation:

Graphical user interface, text, application

Description automatically generated

save and load machine learning models

* Machine learning models often take hours or days to run, especially on large datasets with many features. If your machine goes off, you’ll lose your model and you’ll need to re-train it from scratch.
* Pickle is a useful Python tool that allows you to save your models, to minimize lengthy re-training and allow you to share, commit, and re-load pre-trained machine learning models.

### Save the model:

### To save the model all we need to do is pass the model object into the dump() function of Pickle. This will serialize the object and convert it into a “byte stream” that we can save as a file called model.pkl. You can then store, or [commit to Git](https://practicaldatascience.co.uk/data-science/how-to-use-git-for-your-data-science-projects), this model and run it on unseen test data without the need to re-train the model again from scratch.

For example : saving logistic model using pickle

Graphical user interface, text

Description automatically generated

### Load the model:

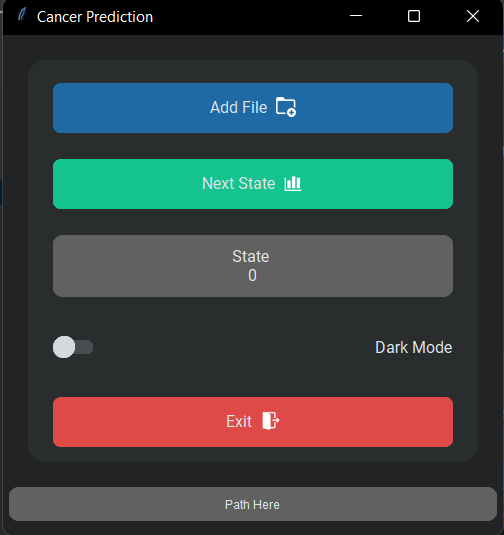
### To load a saved model, all you need to do is pass the “pickled” model into the Pickle load() function and it will be deserialized. By assigning this back to a model object, you can then run your original model’s predict() function, pass in some test data and get back an array of predictions.

Graphical user interface, text

Description automatically generated

GUI for input file using tkinter

The user can upload a test file by using Add file button :



After uploading the test file the result will be malignant or benignant for each sample:

A screenshot of a computer

Description automatically generated with low confidence

* Code Implementation:

Text

Description automatically generated