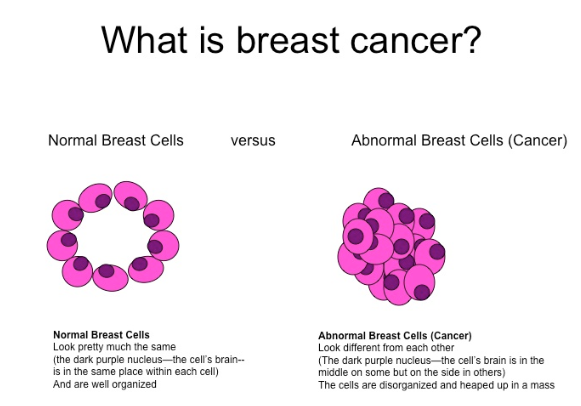
Breast Cancer Prediction Using Machine Learning

Created by: DEEP MAJHI

GitHub Link: <https://github.com/DeepMajhi98/Project-2-Breast-Cancer-Prediction>

Breast cancer is a dangerous disease for women. If it does not identify in the early-stage then the result will be the death of the patient. It is a common cancer in women worldwide. Worldwide near about 12% of women affected by breast cancer and the number is still increasing.

Breast cancer is a disease in which cells in the breast grow out of control. There are different kinds of breast cancer. The kind of breast cancer depends on which cells in the breast turn into cancer.



In this study, we tried to predict Breast Cancer using 4 different Machine Learning Algorithms:

1. Logistic regression classification
2. Decision tree classification
3. SVM (Support Vector Machine) classification
4. Random forest classification

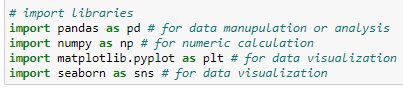
Predictor variable use in classifying breast cancer, its features are computed for each cell nucleus:

1. mean\_radius (mean of distances from center to points on the perimeter)
2. mean\_texture (standard deviation of gray-scale values)
3. mean\_perimeter
4. mean\_area
5. mean\_smoothness (local variation in radius lengths)

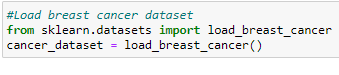
**Project Goal :** We have extracted features of breast cancer patient cells and normal person cells. As a Machine learning engineer / Data Scientist we have to create a ML model to classify malignant and benign tumor.

We measure the study of 4 different algorithms using a **confusion matrix**. And to check which model gives the best Accuracy.

First importing the necessary libraries-



We are using Scikit-Learn inbuilt dataset of Breast Cancer



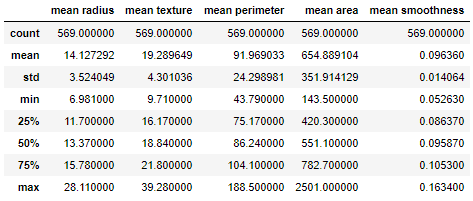
There are total 7 features in the cancer dataset, out of that we are using two features ‘feature\_names’ and ‘target’.

Here ‘features\_names’ is our predictor variable and ‘target’ is the predicted variable. The target variable has two classes – malignant tumor means cancerous cell and benign tumor means non cancerous cell

The dataset which we are using contain 569 rows and 31 columns.

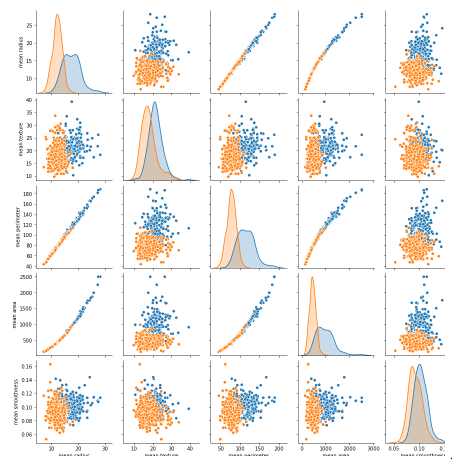
And there is no null-values in the dataset.

I created a statistical distribution of predictor data using .describe() function



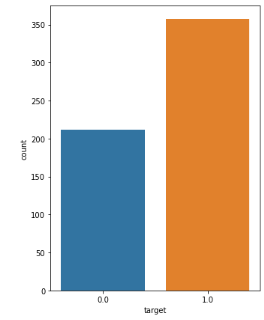
We got the mean, median, maximum, minimum values of each column

For Visualizing the data I first create a pairplot of all the features of predictor data



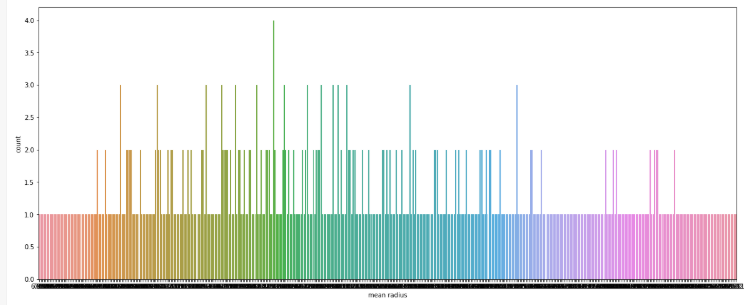
The pair plot showing malignant and benign tumor data distributed in two classes. It is easy to differentiate in the pair plot.

Next I made a Count plot of the target variable Showing the total count of malignant and benign tumor patients in counterplot.

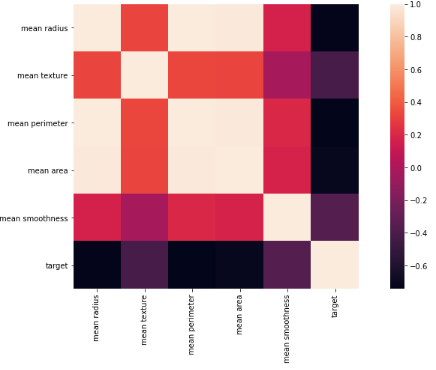


So it counts **Distribution of 357 benign, 212 malignant.**

In the below counterplot max samples mean radius is equal to 1.



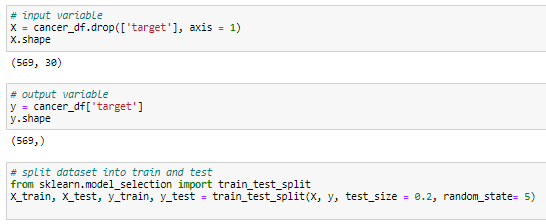
Visualizing the correlations among all the variables-



As you can see above, we obtain the heatmap of correlation among the variables. The color palette in the side represents the amount of correlation among the variables. The lighter shade represents a high correlation

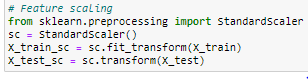
Now comes the data pre-processing part before training the machine learning algorithms.

First we split our data into Train and Test dataset



Data for training and testing To select a set of training data that will be input in the Machine Learning algorithm, to ensure that the classification algorithm training can be generalized well to new data. For this study using a sample size of 20%, assumed it ideal ratio between training and testing.

We Feature Scale the data to Convert different units and magnitude data in one unit using scikit-Learn StandardScaler library.



**Breast Cancer Detection Machine Learning Model Building**

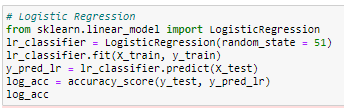
We have clean data to build the Ml model. But which Machine learning algorithm is best for the data we have to find. The output is a categorical format so we will use supervised classification machine learning algorithms.

To build the best model, we have to train and test the dataset with multiple Machine Learning algorithms then we can find the best ML model

**1. Logistic regression classification**

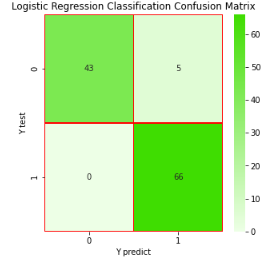
Logistic regression is a technique that can be applied to binary classification problems. This technique uses the logistic function or sigmoid function, which is an S-shaped curve that can assume any real value number and assign it to a value between 0 and 1, but never exactly in those limits. Thus, logistic regression models the probability of the default class (the probability that an input (X)(X) belongs to the default class (Y=1)(Y=1)) (P(X)=P(Y=1|X))(P(X)=P(Y=1|X)). In order to make the prediction of the probability, the logistic function is used, which allows us to obtain the log-odds or the probit. Thus, the model is a linear combination of the inputs, but that this linear combination relates to the log-odds of the default class.

Started from make an instance of the model setting the default values. Specify the inverse of the regularization strength in 10. Trained the logistic regression model with the training data, and then applied such model to the test data.



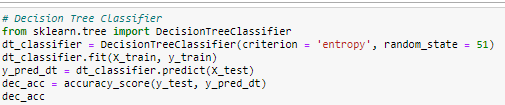
Using logistic regression model we got the accuracy = 0.956140350877193

Confusion Matrix of Logistic Regression –



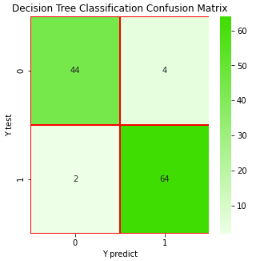
**2. Decision tree classification**

A decision tree is a flowchart-like tree structure where an internal node represents feature, the branch represents a decision rule, and each leaf node represents the outcome. The decision tree analyzes a set of data to construct a set of rules or questions, which are used to predict a class, i.e., the goal of decision tree is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. In this sense the decision tree selects the best attribute using to divide the records, converting that attribute into a decision node and dividing the data set into smaller subsets, to finally start the construction of the tree repeating this process recursively.



Using Decision Tree Classifier model we got the accuracy = 0.9473684210526315

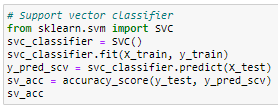
Confusion Matrix of Decision Tree –



**3. SVM (Support Vector Machine) classification**

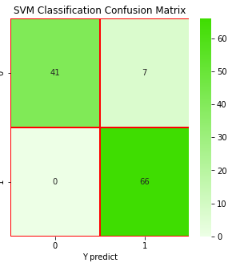
SVMs (Support Vector Machine) have shown a rapid proliferation during the last years. The learning problem setting for SVMs corresponds to a some unknown and nonlinear dependency (mapping, function) y=f(x)y=f(x) between some high-dimensional input vector xx and scalar output yy. It is noteworthy that there is no information on the joint probability functions, therefore, a free distribution learning must be carried out. The only information available is a training data set D=(xi,yi)∈X×Y,i=1D=(xi,yi)∈X×Y,i=1, ll, where ll stands for the number of the training data pairs and is therefore equal to the size of the training data set DD, additionally, yiyi is denoted as didi, where dd stands for a desired (target) value. Hence, SVMs belong to the supervised learning techniques.

From the classification approach, the goal of SVM is to find a hyperplane in an N-dimensional space that clearly classifies the data points. Thus hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes.



Using Support Vector Machine Classifier model we got the accuracy = 0.9385964912280702

Confusion Matrix of Support Vector Machine –

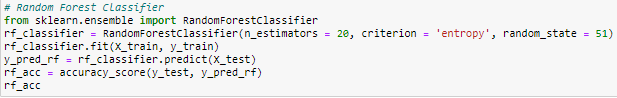


**5. Random forest classification**

Based on the previous classification method, random forest is a supervised learning algorithm that creates a forest randomly. This forest, is a set of decision trees, most of the times trained with the bagging method. The essential idea of bagging is to average many noisy but approximately impartial models, and therefore reduce the variation. Each tree is constructed using the following algorithm:

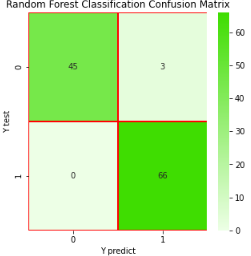
* Let NN be the number of test cases, MM is the number of variables in the classifier.
* Let mm be the number of input variables to be used to determine the decision in a given node; m<Mm<M.
* Choose a training set for this tree and use the rest of the test cases to estimate the error.
* For each node of the tree, randomly choose mm variables on which to base the decision. Calculate the best partition of the training set from the mm variables.

For prediction a new case is pushed down the tree. Then it is assigned the label of the terminal node where it ends. This process is iterated by all the trees in the assembly, and the label that gets the most incidents is reported as the prediction. We define the number of trees in the forest in 100.

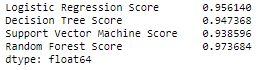


Using Random Forest Classifier model we got the accuracy = 0.9736842105263158

Confusion Matrix of Random Forest –

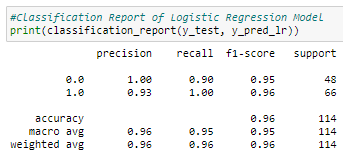


# Comparison of classification techniques

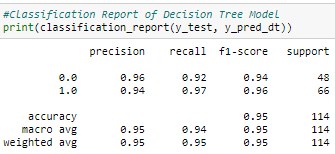


Classification Reports of Models:

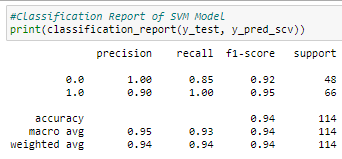
* For Logistic Regression-



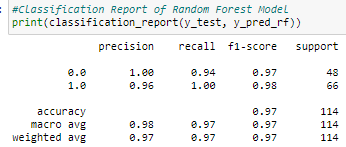
* For Decision Tree:



* For Support Vector Machine:



* For Random Forest:



**Conclusion:**

As we can see above, a Comparison of classification techniques, we can evaluate that here Logistic Regression Classification and Random Forest have the most optimal result of the accuracy.