**Predicting whether a mammogram mass is benign or malignant using different ML algorithms**

# Abstract:

Breast cancer is the leading cause of death among women. There were more than 2 million new cases in 2018 all over the world. In India alone, the rate of breast cancer is 25.8 per 1,00,000 women, which is 87,090 deaths in total. According to WCRF (World Cancer Research Fund), there was strong evidence that consumption of alcoholic drinks, greater birth weight and adult attained height are causes of premenopausal breast cancer. The correct diagnosis of Breast Cancer and the classification of patients into malignant (highly effective) or benign (Not severe) groups is the subject of much research. To predict breast cancer is benign or malignant the machine learning algorithms gives a lot of aid. The algorithms take data from the diagnosis of cancer (BC) and give output as benign or malignant. We will use different ML algorithms like Decision Trees, Random Forest, K- nearest neighbors, Naïve Bayes, SVM and linear-logistic regression for this purpose and try to compare which one excels over the other in different features like runtime, efficiency, etc. The data will be partitioned to training and test data by K- means cross-validation. As support for this approach, a Python-based workflow has been accustomed.

# Keywords:

Breast cancer, malignant and benign, ML, SVM, Random Forest K-nearest neighbours, SVM, regression, cross-validation, python.

# Introduction:

It is believed that the second major cause for death of a woman is due to breast cancer (BC). Usually, BC can easily be detected if specific symptoms appear. But, many women suffering from BC have no symptoms. Hence, regular breast cancer screening is very important for early detection of BC. Detection of breast cancer in early stages increases the chance of survival because it prevents the spreading of malignant cells throughout the entire body. Without the aid of computers it is impossible for health professionals to analyse these complex datasets produced from diagnosis particularly when undertaking complex interrogations of the data. The intelligent healthcare system is therefore a valuable and important domain. AI can be applied to improve BC detection and diagnosis as well as prevent overtreatment. Nevertheless, AI or Machine Learning (ML) methods enables the prediction and empower accurate decision making.

Mammograms have false positive (high-risk) results which show abnormal cells that can lead to unnecessary surgeries. Sometimes surgery done to remove lesions reveals that it is benign which is not harmful. This means that the patient will go through unnecessary painful and expensive surgery. Machine Learning techniques are a huge success in intelligent healthcare systems because of their accuracy. However, it is better if you predict output using accurate methods and big data and learn from the past. This is why our paper is based on Mammogram detection using ML algorithms and comparing the accuracy of several algorithms. If there was no ML technique for BC, physicians would traditionally treat patient based on symptoms. There is no guarantee for accuracy.

We'll be using the "mammographic masses" public dataset from the UCI repository (source: <https://archive.ics.uci.edu/ml/datasets/Mammographic+Mass>)

Original owners of this public database is Prof. Dr. Rüdiger Schulz from Institute of Radiology.

This data contains 961 instances of masses detected in mammograms, and contains the following attributes:

1. BI-RADS assessment: 1 to 5 (ordinal)
2. Age: patient's age in years (integer)
3. Shape: mass shape: round=1 oval=2 lobular=3 irregular=4 (nominal)
4. Margin: mass margin: circumscribed=1 microlobulated=2 obscured=3 ill-defined=4 spiculated=5 (nominal)
5. Density: mass density high=1 iso=2 low=3 fat-containing=4 (ordinal)
6. Severity: benign=0 or malignant=1 (binomial)

BI-RADS is an assessment of how confident the severity classification is; it is not a "predictive" attribute. The age, shape, margin, and density attributes are the features that we will build our model with, and "severity" is the classification we will attempt to predict based on those attributes. Although "shape" and "margin" are nominal data types, which sklearn library for ML typically doesn't deal with well, they are close enough to ordinal that we shouldn't just discard them. The "shape" for example is ordered increasingly from round to irregular.

The data needs to be cleaned; many rows contain missing data, and there may be erroneous data identifiable as outliers as well.

We will apply several different supervised machine learning techniques to this data set, and see which one yields the highest accuracy as measured with K-Fold cross validation (K=10) while comparing and understanding the algorithms.

# References:

1. <https://www.sciencedirect.com/science/article/pii/S2001037014000464>
2. <https://www.who.int/cancer/detection/breastcancer/en/>

SVM

In this algorithm, the data points are considered as standard vectors in a plane formed of the features. It forms a plane with the number of dimensions as the number of features in the data. The new value is predicted based on in which plane it lies similarly like K-means clustering but for higher dimension. SVM has a feature to ignore outliers and find the hyper-plane that has maximum margin explained in detail in kernel parameter below.

Pros: It is effective for data with high dimension. It is useful for both hard margin and soft margin data, the only thing to do is to come up with the optimal penalty variable C.  Due to its nature of Convex Optimization, the solution is global minimum not local minimum. It can be used in a dataset where some of the data is labelled and some are not.

Cons: It doesn’t perform well when we have large data set because the required training time is higher because it works on higher dimensions. However, it performs fast when data is small and has more features.

Parameters:

1. Kernel: There is no perfect shape that can fit in any kind of data given to it uniformly. Hence, the kernel parameter in SVM decides the shape of SVM hyper plane (output). In some cases, there are outliers present in our data which can’t fit in our kernel. SVM can’t deal with outliers correctly because, to maintain a hyper plane shape it has to keep high margin (gap) between the data classes so that it gets good precision in classifying the data that is near to the plane. But as the outliers will never be close their class, they may be on the other partition of the hyper plane and SVM will not care about them. Kernels can have values like linear, polynomial, sigmoid, RBF (Gaussian Radial Basis Function), etc. Kernel function can be written as K(x)=1 if |x|<=1 and 0 otherwise.
2. Gamma: Higher the value of gamma, the hyper plane will try to fit the data classes as exactly as it can and hence it is bad for predicting new values (overfitting). Vice versa, if gamma is low the accuracy of hyper plane will not be high.
3. C: Higher C does mean that the boundary is not going to be smooth which covers more points, but the margin will not be broad. Vice versa for lower C value.

**Decision Tree**

Decision trees are a type of method which uses both classification and regression. Trees answer sequential questions and behave with “if this than that” conditions which finally drive us to the correct decision.

Pros:

1. Easy to interpret and make for straightforward visualizations.
2. The internal workings are capable of being observed and thus make it possible to reproduce work.
3. Can handle both numerical and categorical data.
4. Perform well on large datasets
5. Are extremely fast

Cons:

1. Building a decision tree requires algorithm capable of determining optimal choice at each node.
2. Decision trees are prone to overfitting, especially when a tree is particularly deep. This is due to the amount of specificity we look at leading to smaller sample of events that meet the previous assumptions. This small sample could lead to unsound conclusions

**Random Forests**

Ideally, we would like to minimize both error due to bias and error due to variance which are given in the cons section of decision tree. Random forests mitigate this problem well. A random forest is simply a collection of decision trees whose results are aggregated into one final result. Their ability to limit overfitting without substantially increasing error due to bias is why they are such powerful models.

Random forest reduce veriance by training differant model and that all model reflects on the final decision of the dataset that we predect. Hence, random forest is uch more robust metho than simple decision tree.

**Recall:** Recall gives us an idea about when it’s actually yes, how often does it predict yes.

Recall=TP / (TP + FN)=100/(100+5)=0.95

**Precision:** Precsion tells us about when it predicts yes, how often is it correct.

Precision = TP / (TP + FP)=100/ (100+10)=0.91

Accuracy=(TP+TN)/(TP+TN+FP+FN)

# Sensitivity, hit rate, recall, or true positive rate

TPR = TP/(TP+FN)

# Specificity or true negative rate

TNR = TN/(TN+FP)

# Fall out or false positive rate

FPR = FP/(FP+TN)

# False negative rate

FNR = FN/(TP+FN)

# Precision or positive predictive value

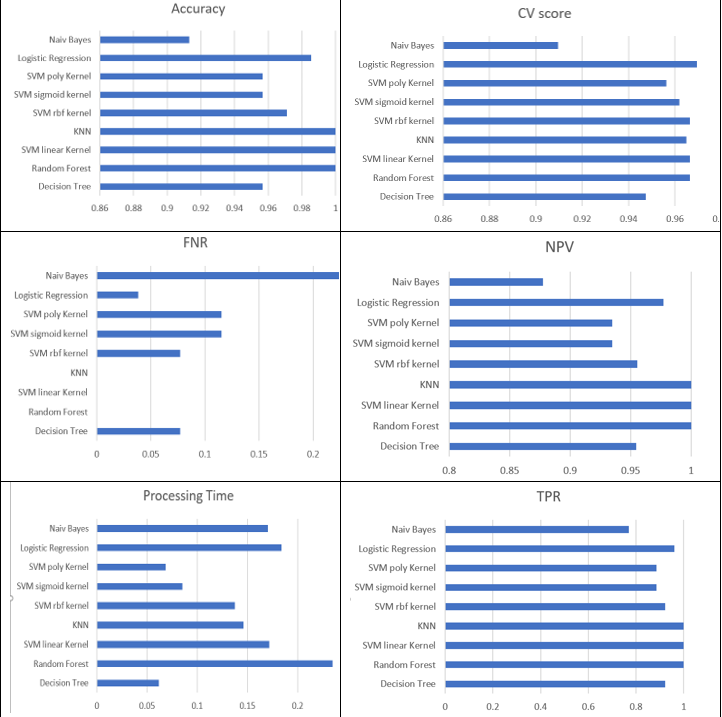
PPV = TP/(TP+FP)

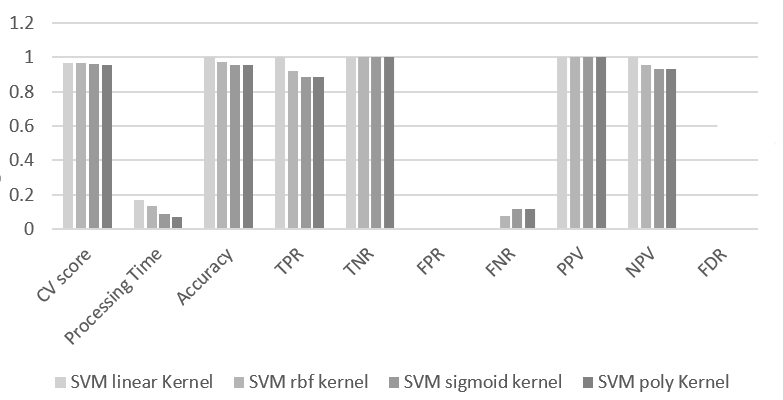
# Negative predictive value

NPV = TN/(TN+FN)

# False discovery rate

FDR = FP/(TP+FP)



Figures above shows differant results according to differant kernels in SVM algorithms,

KNN

K-nearest neighbours simply calculates the nearest K points to our concerned (new) data point through Euclidian distance and compares if the data(characteristics) of this new point is similar to the data of those K-points. We may think that higher the k, more accurate the data is, but it is not true. If K is high, algorithm will compare new point to those points that are far too and the answer accuracy will decrease. Vice Versa for the case when K is small. The Euclidian distance between p and q will be calculated as under:

