Differentiation of Malignant and Benign Breast Lesions Using Machine Learning Algorithms

Chetan Nashte, Jagannath Nalavade, Abhilash Darvemula, Seema Singh, Meghana Bhowmick
Department of Computer Engineering, Sinhgad Institute of Technology, Lonavala, India

***Abstract* — Medical diagnosis is a process which requires critical decisions to be made by a medical professional. These decisions can be made using a Clinical Decision Support System (CDSS) to speedup or assist the decision making process. Many Machine Learning algorithms have been used in a CDSS to make accurate predictions. These systems improve themselves with the increasing number of data that comes in. This paper explores the possibility of using supervised classification algorithm in a CDSS to predict whether the breast lesion is malignant or benign. Various machine learning algorithms have been implemented to know which would be a suitable match for use in a CDSS. We used a dataset with characteristics of cell nuclei and analyzed it with statistical tools to find out the most significant features that help to better classify it. In our experiment the kernel based method called Support Vector Machine had the best performance with an accuracy score of 96% making it an ideal implementation for classifying benign and malignant breast lesions.**

Keywords— Clinical Decision Support System, Medical Diagnosis, Breast Cancer, Support Vector Machines, Random Forest, Multilayer Perceptron, Decision Trees.

I. Introduction

The CDSS discussed here is a non-knowledge based CDSS that use machine learning to process previous data records of patients and form a generalized inference out of it [1]. A review made in 2014 did not reveal any significant benefit in terms of risk of death by the use of CDSS [2]. This has shown a huge scope for research in developing a successful system. Judging by the potentials of what Artificial Intelligence can do, it would be a wise act to dream of a CDSS which would outperform human capability to deliver astute diagnosis. The hurdles like data acquisition and using the right algorithms to process the data are the ones slowing down the development of CDSS. There have been many attempts to make the datasets publicly available like the 'UCI Machine Learning Repository'. This has been very helpful and has accelerated the research in this area to some extent.

There has been a great deal of research in aiding cancer diagnosis using computers. Jiang et al. [3] developed an automated computer scheme that was demonstrated to classify clustered micro calcifications more accurately than radiologists. Breast cancer has become the biggest cause for concern among women diagnosed with cancer. Physicians usually diagnose breast cancer by physically examining the tumors and classifying them into benign or malignant. This task heavily relies on the reliability of the physician's skill. Any wrong diagnosis can lead to a healthy patient going for a cancer treatment. Using a computer aided diagnosis system can greatly reduce the chance of wrong diagnosis.

II. Description of the methodology and algorithms

We have used supervised machine learning algorithms to process the dataset obtained from the UCI repository. The dataset used here has features computed from a digitized images of a Fine Needle Aspirate (FNA). The features are characteristics of the cell nuclei such as its radius, texture, etc. It has five hundred and seventy instances with thirty-one features. Each instance has its class marked as malignant or benign.

For any classifier to perform at its best, the features in the dataset needs to be carefully selected. Not all the features from the dataset may lead to an accurate classification. So we have used statistical methods to analyze the dataset and further refine the use.A novel approach was proposed by Hosseeinzadeh et al. [1] which uses up-up keystroke latency feature. In comparison with the prevalent key down-down and hold-down features, the up-up keystroke latency (UUKL) features proved to be more beneficial. The comparison was done using a GMM based verification system.

A. *Principal Component Analysis*

PCA is a very important procedure to reduce the dimension of the data [4]. We have used PCA to select the most relevant features and reduce the dataset to a lower dimension. This has helped to eliminate most of the noise and lowered the complexity of the data. PCA aggregates highly correlated features together.

B. *Cross-Validation*

Classification algorithms learn by modifying their parameters at each iteration. A model that learns on a specific data will have a good prediction score if tested on the same set of data but would have a significant chance of failing on a data it has never seen. So it is a good practice to divide the given dataset into three categories:

- Training set
- Validation set
- Test set

The training set is used to train the model by finding the most optimal parameters. While the validation set is used to tune the hyperparameter like the learning rate in the logistic regression model. The test set is used to assess the performance of the classifier and find out the error score.

C. Algorithms

- Naive Bayes

It is a probabilistic classification technique that relies on the assumption that the features are independent of each other. It selects the most probable hypothesis (h) given an instance of data (d). Bayes theorem is used to calculate the probability of hypothesis given a prior knowledge. The algorithm has no free parameters to be set unlike Neural Network and Support Vector Machines [9].

$$P(h|d) = (P(d|h) * P(h)) / P(d) \qquad (1)$$

Where,

P(h|d) is the probability of hypothesis h given the data d, which is the posterior probability.

P(d|h) is the probability of data d provided that the hypothesis h was true.

P(h) is the probability of hypothesis h being true (regardless of the data), which is the prior probability of h.

P(d) is the probability of the data.

To predict the class, the hypothesis with the highest probability is selected after calculating the posterior probability for different hypothesis.

- K Nearest Neighbor (KNN)

KNN is widely used for classification as well as regression problems. It is a non-parametric method. In order to carry out prediction, it requires all of the training dataset to predict the class of the test data. It does not do any generalization and hence has no or minimal training phase. It assumes that the dataset is a feature space. The neighboring points are ranked inversely with respect to the test data point in the multi-dimensional space [6]. The number k decides how many neighbors to consider for classification. The class of the test data point is decided based on the class that has major points in the k number of nearest neighbors. The performance of this algorithm varies with the size of data [7, 8].

- Decision Trees and Random Forest

Decision Tree is a supervised algorithm that is mostly used for classification. It segregates the dataset based on all its feature values and then decides which creates a more homogeneous set. It uses strategies like Gini Index, Chi-Square, etc. to decide how to split the node [10]. Decision Trees are better at dealing with noise, missing values, and redundant features. Although the algorithms are a bit difficult when it comes to handling high dimensional data. The errors generated in the tree creates a problem as the number of classes increase [11, 12].

Random Forest is an ensemble method that trains number of decision tree classifiers over many sub-sets of dataset and outputs the class that appears the most [13]. In case of regression, the output is the mean response from all the trees. It uses feature bagging which is good way of reducing variance in the dataset. A random subset of features is given to the tree at each node to pick from that subset instead of the whole set. This increases the randomness in the system. Random Forests are very popular for classification and in many case outperforms the Support Vector Machines.

- Support Vector Machine

SVM's can be used for both classification and regression. Here the algorithm plots all the data into an n-dimensional hyperspace and tries to classify it by having a hyperplane that separates the classes. It is based on maximizing the minimum distance between the nearest data point and the hyperplane [14]. The accuracy and performance here are independent of the size of data and dependent on the number of training cycles.

- Multilayer Perceptron

MLP is a feedforward neural network that maps set of input data onto a set of output data. It consists an input layer, hidden layer, and an output layer. MLP can easily classify data that is linearly not separable. A multilayered perceptron is Each unit in a particular layer is connected to all the units of the next layer. Each unit is bounded by an activation function and each layer has its own set of parameters except the input layer. The algorithm tries to minimize the cost function at every layer by altering the weights. It is robust in handling irrelevant input and noise [15]. It is very crucial to decide the number of units in the hidden layer as underestimation can cause poor approximation and overestimation can cause over-fitting.

III. EXPERIMENTS

A. Principal Component Analysis

Principal Component Analysis will state the variance in each component and the weights associated to each feature. Table. I, shows the values of PCA on the dataset. The dimension 1 has 98.20% of variance. SO all the variance in the data can be explained by one dimension. In dimension 1, 'Area mean' and 'Area worst' are the ones with most of the feature weights.

B. Feature Selection

Having high dimensional data can sometimes add to lower prediction score. It is necessary to select features carefully to have the optimal results. We plotted a histogram based on their labels for each feature that had a significant value in our principal component analysis.

TABLE I.

	Dimension 1	**Dimension 2**	**Dimension 3**
Explained Variance	0.9820	0.0162	0.0016
Radius mean	0.0051	0.0093	-0.0123
Texture mean	0.0022	-0.0029	-0.0064
Perimeter mean	0.0351	0.0627	-0.0717
Area mean	0.5168	0.8518	-0.0279
Smoothness mean	0.0000	-0.0000	0.0001
Compactness mean	0.0000	-0.0000	0.0001
Concavity mean	0.0001	0.0001	0.0003
Concave points mean	0.0	0.0	0.0
Symmetry mean	0.0000	-0.0000	0.0001
Radius worst	0.0072	-0.0006	-0.0156
Texture worst	0.0031	-0.0132	-0.0315
Perimeter worst	0.0495	-0.0002	-0.0923
Area worst	0.8521	-0.5197	-0.0393
Smoothness worst	0.0000	-0.0001	-0.0000
Compactness worst	0.0001	-0.0003	-0.0008
Concavity worst	0.0002	-0.0002	-0.0008
Concave points worst	0.0001	-0.0000	-0.0003
Symmetry worst	0.0000	-0.0002	-0.0003
Fractal dimension worst	0.0000	-0.0001	-0.0000

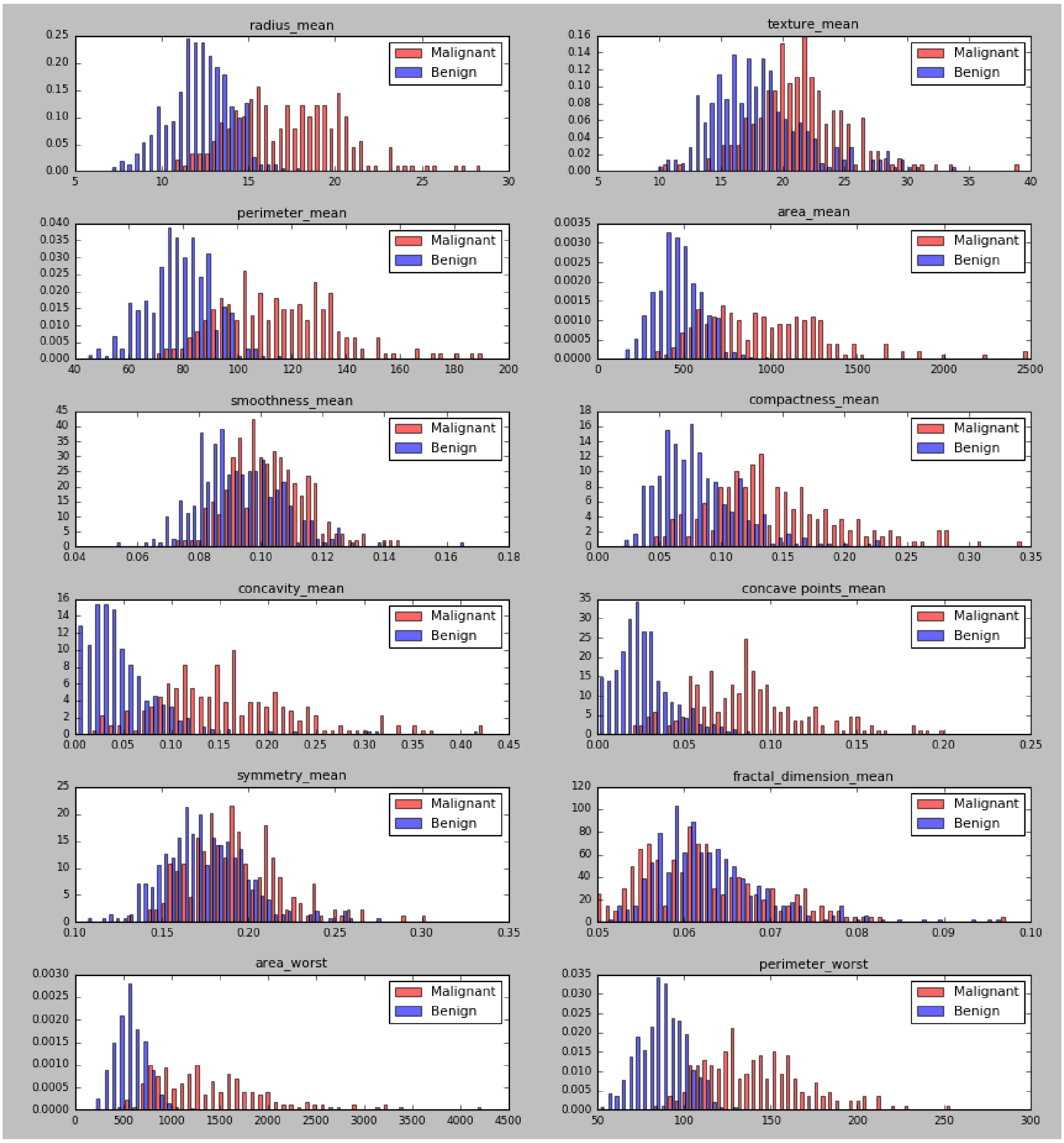

Fig. 1. *Histogram plot*

Fig. 1, shows the graph of histograms. From the above graph we can see that the radius mean, perimeter mean, area mean, concavity mean, concave points mean, area worst and perimeter worst have distinct grouping between benign and malignant type. These are good candidates for selecting them in the feature set. To further analyze the features, we plotted a scatter matrix by plotting each feature with the other features. This gave a more detailed view of correlation between features. Fig. 2 shows scatter matrix. The perimeter mean, area mean and radius mean show strong correlation.

C. Training and validation phase

The features selected in the feature selection phase were used to train the following classification algorithms.

- Naïve Bayes
- Decision Tree
- Random Forest
- Multilayer Perceptron
- Support Vector Machine
- k-Nearest Neighbor

All the models were trained for various hyperparameters to get the optimum results. 5-fold and 10-fold validation techniques were used to cross validate the data.

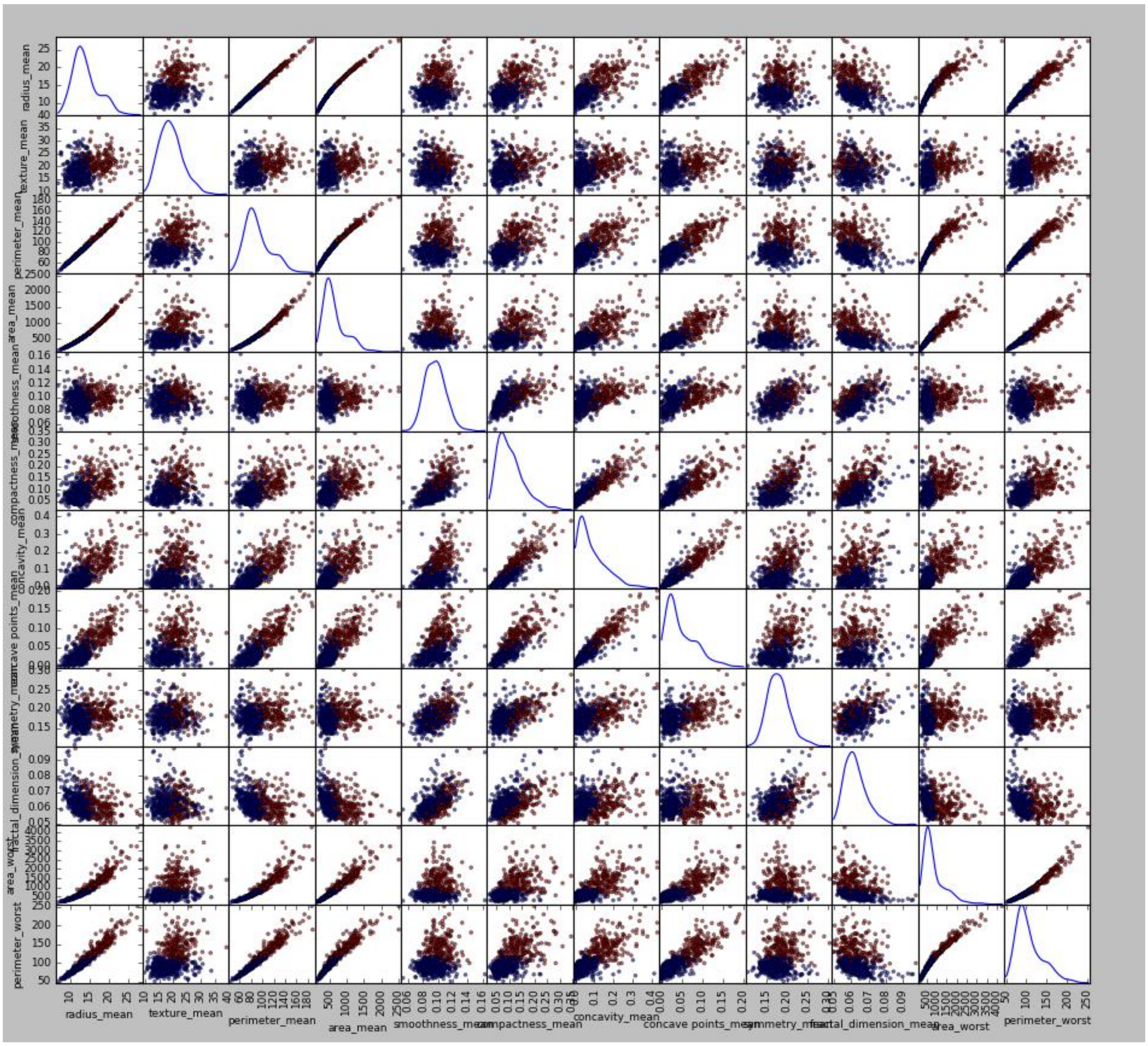

Fig. 2. Scatter matrix plot

IV. Results

Table. II shows the accuracy scores to predict correct class for an instance. The multilayer perceptron has performed very poorly with a score of just 0.72. It had nine input layer, nine hidden layers and one output layers. The layers were kept equal to the number of features except the output layer. The algorithm took relatively high iteration to converge. At about half of the iteration cycle the algorithm had a good score of 0.87 and there on kept reducing. This clearly shows over-fitting behavior. Support Vector machine outperformed all other algorithm. The SVM used Linear kernel and Radial Basis Function of which Linear Kernel gave a better score.

TABLE. II

Sr. No.	Classifier	Accuracy Score (Cross Validation)
1.	Naive Bayes	0.92
2.	k-Nearest Neighbor	0.93
3.	Decision Tree	0.95
4.	Support Vector Machine	0.96
5.	Random Forest	0.95
6.	Multilayer Perceptron	0.72

V. Conclusion

The best model to differentiate between benign and malignant breast lesions seems to be Support Vector Machine. Decision tree and Random forest take second place and are also a good choice if low dimensional dataset is to be used. Neural network does not seem to be a good option for smaller datasets. SVM is a good candidate for use in the Clinical Decision Support System. With further research efforts like more data acquisition and better learning algorithms, CDSS can aid in accelerating the process of diagnosis and hence prove to be a useful tool in saving more lives.

VI. References

[1] "Tanveer Syeda-Mahmood plenary talk: The Role of Machine Learning in Clinical Decision Support". SPIE Newsroom. March 2015.

[2] Moja, L; Kwag, KH; Lytras, T; Bertizzolo, L; Brandt, L; Pecoraro, V; Rigon, G; Vaona, A; Ruggiero, F; Mangia, M; Iorio, A; Kunnamo, I; Bonovas, S, "Effectiveness of computerized decision support systems linked to electronic health records: a systematic review and meta-analysis," American Journal of Public Health, December 2014.

[3] Y. Jiang, R. M. Nishikawa, E. E.Wolverton, C. E. Metz, M. L. Giger, R. A. Schmidt, and C. J. Vyborny, "Malignant and benign clustered microcalcifications: automated feature analysis and classification," Radiology, vol. 198, pp. 671–678, 1996.

[4] Ilin, Alexander, Raiko, Tapani, "Practical approaches to principal component analysis in the presence of missing values," J. Mach. Learn. Res. 11, 2010.

[5] Mehmet Fatih Akay, "Support vector machines combined with feature selection for breast cancer diagnosis," Expert Systems with Application, 2009.

[6] A. C. Lorean et al., "Comparing machine learning classifiers in potential distribution modelling," Expert System with Applications, vol. 38, pp. 5268-5275, 2011.

[7] F. Pernkopf, "Bayesian network classifiers versus selective k-NN classifiers," Pattern Recognition, vol. 38, no. 1, pp. 1010, 2005.

[8] M. J. Islam, Q. M. J. Wu, M. Ahmadi, and M. A. Sid-Ahmed, "Investigating the performance of Naive-Bayes classifiers and K-NN classifiers," Journal of Convergence Information Technology, vol. 5, no. 2, 2010.

[9] L. I. Kuncheva, "On the optimality of Naive Bayes with dependent binary features," Pattern Recognition Letters, vol. 27, pp.830-837, 2006.

[10] L. Rokach and O. Maimon, "Top-Down induction of decision trees classifiers-a survey," IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews, vol. 35, no.4, pp. 476-487, 2005.

[11] C. J. Hinde, and R. G. Stone D. Xhemali, "Naive Bayes vs Decision Trees Vs Neural Networks in the Classification of Training Web Pages, "International Journal of Computer Science Issue, vol. 4, no.1, 2009.

[12] N. B. Amor, S. Benferhat, and Z. Elouedi," Naive bayes vs decision trees in intrusion detection systems," in ACM symposium on Applied computing, pp.420-424, 2004.

[13] A. C. Lorean et al., "Computing machine learning classifiers in potential distribution modelling," Experts Systems with Applications, vol. 38, pp. 5268-5275, 2011.

[14] M. Aly, "Survey on multiclass classification methods," Neural Network, pp. 1-9, 2005.

[15] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification," Informatica (Slovenia) vol. 31, pp. 249-268, 2007.