Breast Cancer Detection

Aditi Malladi
Department of Computer and Information Science and Engineering
University of Florida
aditi.malladi@ufl.edu

***Abstract*—Breast Cancer is a silently growing epidemic claiming thousands of lives every year. In the initial stages no, real symptoms are displayed and therefore multitudes of people are diagnosed with this kind of cancer, and more often than not the discovery is at a late stage due to back logged medical systems and high prices of tests. Rapid and frequent screening is not possible due to these reasons. Building a classification technique to distinguish Malignant Breast Cancer from Benign Breast Cancer can help tackle this widespread issue. It will help circumvent the need to wait for biopsy results once a growth has been discovered. This study is aimed at comparing multiple classification techniques to achieve the highest possible accuracy in labeling growths as Benign or Malignant. The various methods being compared are Random Forest Classifier, k-Nearest Neighbors, Naive Bayes, Logistic Regression, Support Vector Machines and implementation of Artificial Neural Networks (ANNs).**

***Index Terms*—Breast Cancer Detection, Mammary Carcinoma, Tumor Classification, Comparison of Classification Models, Cancer Diagnosis, Ductal Carcinoma In Situ (DCIS)**

I. Introduction

BREAST cancer is a growing concern, amongst the various detected cancer found worldwide. It is caused by a group of cells that grow and rapidly mutate. Many types of cancer cells finally form lumps or massed called a tumors and are named after the part of the body where the tumor originates. "2019, an estimated 268,600 new cases of invasive breast cancer were diagnosed among women and approximately 2,670 cases were diagnosed in men. In addition, an estimated 48,100 cases of DCIS or Ductal Carcinoma In-Situ will be diagnosed among women. Approximately 41,760 women and 500 men are expected to die from breast cancer in 2019" [3]. Breast cancer typically produces no symptoms when the tumor is small and most easily cured [3]. This proves that if multiple cancer screenings were done early on and regularly this can be caught and cured. But like any other medical problem, the costs of doing multiple screenings is exponential, time consuming and such resources aren't affordable by the masses, both in terms of time and monetary concerns. Using the help of innovative software can hasten the diagnosis process by decreasing the amount of input required by a highly qualified medical professional during the entire process. Having an effective software to diagnose cancer or even flag potential patients would help thousands of people receive medical attention and also reduce the number of patients and strain to the health care system by eliminating those who do not potential cancerous features or cells. Research in this area can be done by leveraging the vast quantum of medical data available. This would in turn also reduce the medical cost per person and help reach more people and thereby conduct more screenings each year. Bringing in automatic diagnostic tools would also help eliminate some of the human error encountered. This report makes a detailed examination to generate a system to distinguish tumor growths from Malignant and Benign and clearly classify them. This is aimed to be done by comparing the results from multiple classification techniques namely, Naive Bayes, Random Forest Classifier, k-Nearest Neighbors, Logistic Regression, Support Vector Machine and ANNs time again have been used for multiple classification tasks resulting is remarkable performance, making them an ideal choice in this study. A Malignant tumor are categorized as cancerous growths that often resist medications and can invade nearby tissues. These rapidly multiplying cells can also travel to other parts of the body via the blood stream, potentially affecting more areas of the body, making a timely diagnosis more essential. Breast Cancer starting in the Breast tissue can spread to the liver or bones (and other parts of the body). Generally, only a Biopsy can make a classification on Malignant or Benign growths. Achieving high classification results can help eliminate this step and hasten the process of discovery, helping a timely diagnosis. A Benign tumor on the other hand is not cancerous and won't metastasize. It becomes a problem only if the growth affects nearby organs, vessels or nerves. These tumors can grow to be very large and often have to be removed surgically.

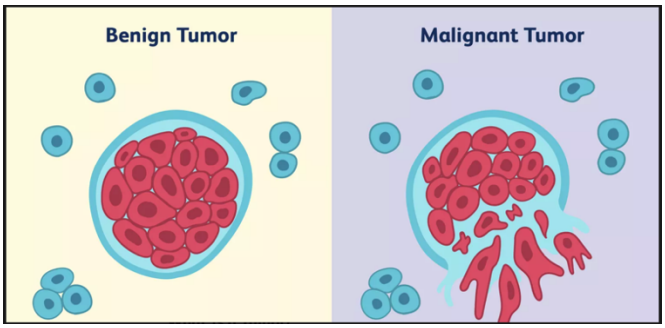

Figure 1 : Difference between Malignant and Benign Tumors [6]

All the mentioned classifications have been run and the predicted outcomes and the computed accuracy for each model, have been quantified by doing an in-depth comparative analysis on the various results to be able to tell which models are best able to make the most accurate predictions. These results have been summarized in the Section IV under Experiments and Results.

What makes cancer diagnosis or detection an interesting topic is not just the benefits of the successful implementation but also the intricacy of the data and the relationships between the data features and the outcomes. To classify a given Carcinoma as Benign or Malignant depends on multiple factors some more than others. This research would help understand the different features available in cancer studies and also rank the more important features which would help medical professionals focus on those more in the upcoming research work.

II. Related Work

There has a lot work been done in the prediction of Cancer visa Medical Images and also via Genetic Studies. The problem in these aspects are medical images for specific learning methods are not widely available and are generally not open source. This causes a lack of training data and hence would not yield the most optimal results.

Genetic Profiling comes with its own drawbacks. The first one being the very expensive price. Even though in the recent years there has been a significant reduction in price aspect, it still requires significant resources, compute power and storage requirements as this study generates hundreds of TB of information. A gene expression profiling method to understand the underlying features of Breast Cancer has been studied, but this still leaves a dearth in the Physical Diagnosis of the cancer.

This study claims to focus classification techniques for Benign and Malignant cancer detection, thereby following a earlier diagnosis of a tumorous growth in the patient. This is study is focused on achieving the best possible accuracy after a potential patient has a abnormal growth detected. Such studies can also be extended to other cancers once the features have been thoroughly understood.

III. Methodology and Materials

As described in the Project Proposal, a step by step approach has been adopted in understanding the dataset, exploring the features and then running suitable methods of classification. Multiple steps, methods of classification, learning and tools have been experimented with which have been described in this section.

A. Data Visualizations and Feature Extractions & Selection

The first step is to better understand the different features provided in the dataset being used. The UCI Breast Cancer Wisconsin Dataset [7] has features that have already been calculated from Fine Needle Aspirate images of cancer. From a superficial perusal of the dataset it is evident that the first column, that is the ID cannot be used for classification. The list of the features included in the Wisconsin Dataset are as shown in Figure 2.

```
Index(['id', 'diagnosis', 'radius_mean"', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst', 'Unnamed: 32'],
      dtype='object')
```

Figure 2: Features of Dataset

As we can see there is an Unnamed Feature column listed as well, which we can exclude for our analysis for now. Now from the Figure 3 we can see how the data is distributed into Malignant and Benign forms of cancers. This information helps us understand how the dataset is distributed.

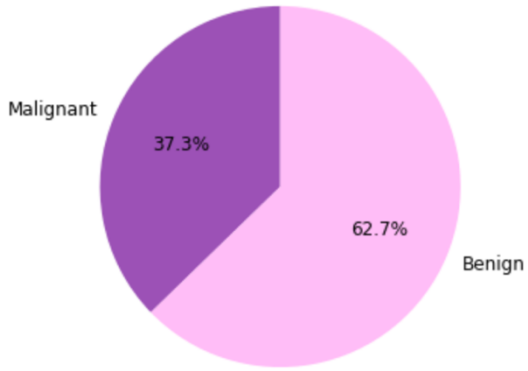

Figure 3: Distribution of Malignant and Benign outcomes in the Dataset

Moving forward it is essential to now understand which features give a clear classification into Malignant or Benign. This can be done by using a few different visualization techniques to help us understand the distribution across the features. This in turn helps us pick features to use for various classification methods. The first step is a simple violin plot is implemented using the Seaborn Package. And to affirm the findings from the first plots done, a second round of swarm plots is done to make sure the findings are consistent as well and meaningful classification from the said features can be performed.

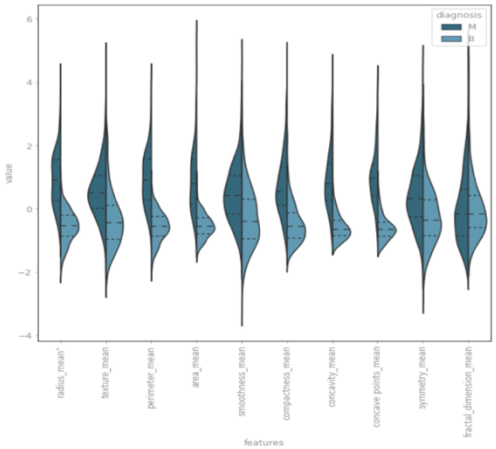

Figure 4: Violin Plot of dataset - 1/3

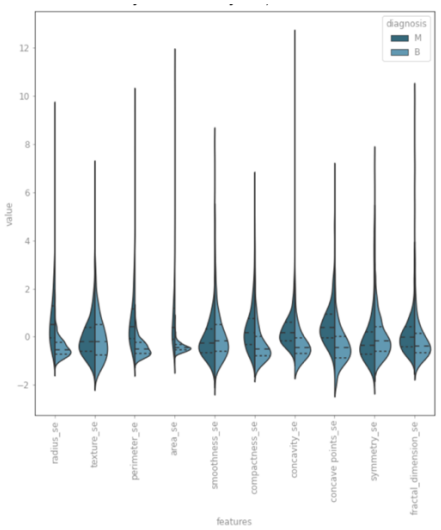

Figure 5: Violin Plot of dataset - 2/3

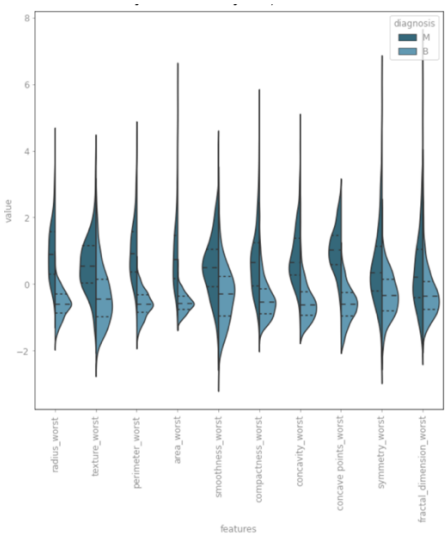

Figure 6: Violin Plot of dataset - 3/3

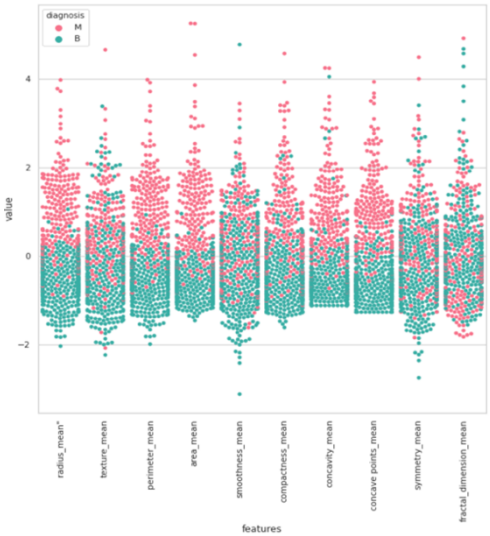

Figure 7: Swarm Plot of dataset - 1/3

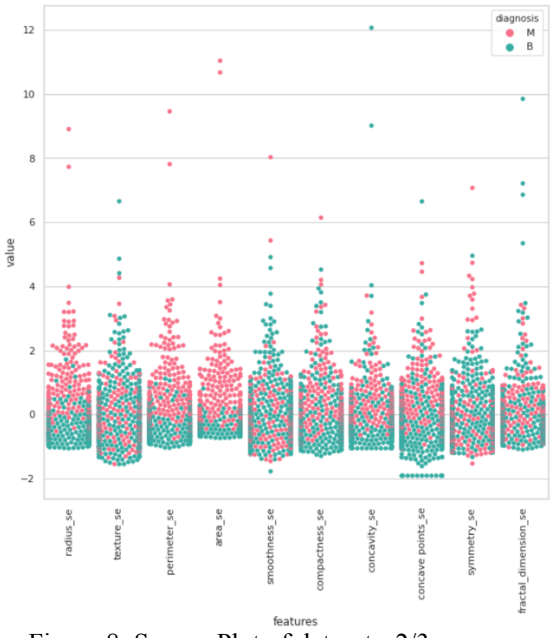

Figure 8: Swarm Plot of dataset - 2/3

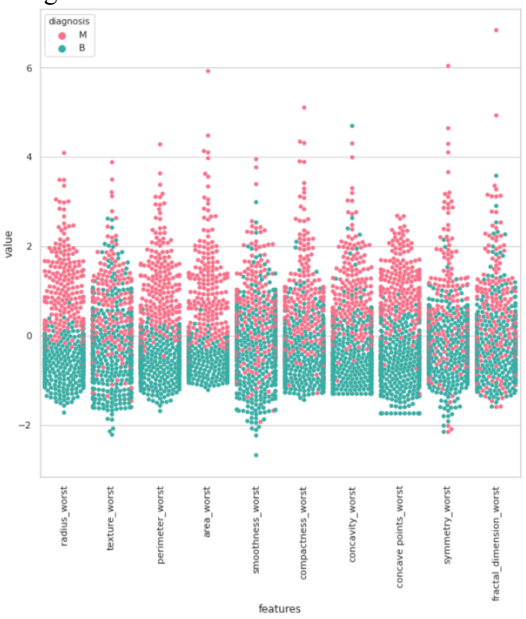

Figure 9: Swarm Plot of dataset - 3/3

From the above plots the following can be interpreted. The mean texture data seems classified between Malignant and Benign and would make a very good selected feature for classification purposes. Fractal Mean on the other hand from just looking at the violin plot does not give any useful information, as it is not separated data, making is an ill-suited feature for classification. From the plots concavity mean and worst also seem to produce very similar results. Being similar allows us to drop one of the features as they would essentially not make a difference in the classification outcomes. The data found under area – worst gives us a very clear separation between malignant and benign data for the most part. Smoothness – se on the other hand is completely mingled and hence would make a bad choice as a feature for classification.

B. *Naïve Bayes Classification*

The Naive Bayes classifier is a very widely used classifier based on the Bayes Theorem. It is a simple and easy to use model, even for really large datasets. This method is particularly great for when the features in the dataset are independent or unrelated to one and other. The algorithm can be tested in this study as based on the given seemingly independent features (or loosely dependent features). The idea here is to train a conditional probability model give as follows: $\mathbf{p(C_k|\ x_1,\ x_2,\ \ldots\ldots,\ x_k)}$. Here, they could be **k** different classes. For this study k = 2. The Bayes theorem is as follows**:** $\mathbf{p(C_k|x)=p(C_k)\ /\ p(x|C_k)p(x)}$.

C. *Random Forest Classifier*

One of the implementations explores the classification technique of the Random Forest Classifier. It is an ensemble algorithm. It created a set of decision trees from a randomly picked sub set of training data. Then, it averages over the outcomes from the various trees to place the test object in the final class. Sklearn machine learning package in python provides this function. This is a particularly popular classifier as it uses multiple trees. Using a single tree can lead to overfitting of the training data, while in this method the samples are drawn randomly with replacement, ensuring that each one will be used frequently. Each decision tree will be trained on different samples from the training data and overall the entire "forest" of decision trees will produce a lower variance without the cost of increasing the bias.

D. *Logistic Regression*

This method of implementing Binary Classification is highly useful in computing the response of a test object into one of two responses. It predicts the probability of a result when it can have one of these two responses. Given this dataset and the multiple features that it contains a linear regression would not be a good choice, as multiple features might not linearly fit leading to, too many outliers. Producing a logistic regression would limit the outcomes to lie between 0 and 1. Logistic Regression uses the Maximum Likelihood Estimation to compute the coefficients. This function is run till the log likelihood does not significantly vary. The equation is shown in Figure 10 [8].

$$\boldsymbol{\beta^1 = \beta^0 + [X^T W X]^{-1}.X^T(y-\mu)}$$

$\boldsymbol{\beta}$ *is a vector of the logistic regression coefficients.*

$\boldsymbol{W}$ *is a square matrix of order N with elements* $n_i\pi_i(1-\pi_i)$ *on the diagonal and zeros everywhere else.*

$\boldsymbol{\mu}$ *is a vector of length N with elements* $\mu_i = n_i\pi_i$.

Figure 10: Logistic Regression Coefficients

E. *k-Nearest Neighbors*

This is a supervised machine learning algorithm that can be used to implement this classification problem. It leverages the fact that things that are in close proximity are similar or related to each other. This is efficient in the current dataset as the closely related features for a particular outcome give the same results. Multiple different distance metrics can be tested, and the ones used can be seen in the Figure 11 along with their formula [2].

Distance functions

Euclidean $\quad \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$

Manhattan $\quad \sum_{i=1}^{k}|x_i - y_i|$

Minkowski $\quad \left(\sum_{i=1}^{k}(|x_i - y_i|)^q\right)^{1/q}$

Figure 11: Formula for distance metrics

For *k*-NN classification, an input is classified by a majority vote of its neighbours. If k is set to one then the test object is assigned to the outcome of the nearest neighbor. But it is important to note in this technique that before using k-NN it is a practice to standardize the training set before. Parameter selection is also performed to optimize the output, so as to group together the most useful of the features from the dataset together thereby giving the best possible outcome from k-NN.

F. *Support Vector Machines*

This is again a supervised machine learning model that can be executed here given the labelled dataset that we have. Each data element is projected onto a n-dimensional space, based on the number of features that exist in the dataset. After this the model works by computing the hyper plane that separate the two outcomes (in this case Malignant or Benign). The vectors are just the coordinates of the feature data. The problem is taken into the input space dimension and projected into higher dimensional data to compute the hyperplane.

G. *Artificial Neural Networks*

The use of Artificial Neural Networks or ANNs is very widespread. They do not always require a lot of pre-work like feature extraction, but it is still useful to pair along with some feature extraction methods to obtain better results. They help by learning weights to correctly classify and cluster. They comprise of one or more layers of nodes, based on which they can be classified as a Single Layer Perceptron or a Multi-Layer Perceptron. For our classification with the Wisconsin data we will be using the Supervised method of learning as we have labeled data whose outcome we already have generated. Various MLP architecture's results are compared in this study comparing performance of various optimizers, mentioned as follows:

a. Gradient Descent Optimizer: The most popular optimization approach for an ANN. The weight is updated by utilizing the present computed gradient $\mathbf{\delta L/\delta w}$ by multiplying with the learning rate.

b. Adam Optimizer: It is a popular adaptive optimizer used in various deep – learning approaches mainly for the reason that it utilizer the momentum or the computer moving average of the gradient, unlike in the previous optimizer. The moment formula is

given as $\mathbf{m_n = E[X^n]}$ where the moment is given by n of a random variable X.

Various activation functions as well were explored to pick the best possible function for the dataset. The following were tested:

a. ReLU: Rectified Linear Unit, also the most frequently used activation function. The equation of the same is **f(x) = max{0, x}.** For positive values the function behaves like a linear function, when being used while training a neural network using backpropagation. This function returns zero for all negative values.

b. Sigmoid Function: This is a popular choice for activation function as the function returns values between 0 to 1 making it a suitable choice in situations when the probability needs to be computed, making it a suitable function to be tested in this study. The equation is $\mathbf{f(x) = 1/(1 + e^{-x})}$.

c. Hyperbolic Tangent Function: It is similar to the logistic sigmoid function and it's range is between **(-1, 1)**. This is a particularly advantageous function as all values below zero are all mapped near zero in the graph. It is useful for classification problems.

H. Principal Component Analysis

Principal Component Analysis or PCA is a widely used statistical process to reduce feature dimensionality of a given set of features. It is an orthogonal transformation to find out the key or principal components and highlight them while being able to reduce the features which are co-related. The remaining "key" features are understood to make most of the variability of the data. It is often an essential step to run before classification results as it helps avoid features that would have very little effect on the outcome.

IV. Experiments and Results

After reading through and exploring the best methods of classification, multiple experiments were run on the datasets to analyze the outcomes using the best methods. The various findings in the Data Visualization and Feature Extraction methods have been taken into account in the following methods. Various Classification techniques have given varying outcomes as we will see below. It will prove to be useful to run the same classifier multiple time while changing parameters to achieve the best possible outcome.

A. Classification using Naïve Bayes

To implement the Naïve Bayes function the sklearn package in python is used. With what was learnt from the dataset's features from the previous visualizations we use a Gaussian Classifier in our first test to classify the data. The results according to the testing and training set ratio are displayed in Figure 12. From the accuracy data visualization in Figure 12 we can see that using a testing ratio of 0.4 showed a significant improvement in the results giving us an accuracy of 95.18% following a dip in the accuracy when using a ratio of 0.3 and then again showing a very significant increase in the accuracy.

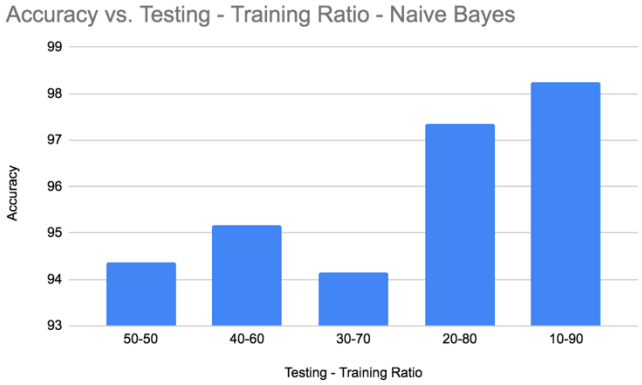

Figure 12: Accuracy (%) vs. Testing-Training Ratio for Naive Bayes

Considering the increment and decrement and increment again, the final accuracy was computed using a set of 0.35 with a final accuracy from Naive Bayes as 95%.

B. Classifications using Random Forest Classifier

Initially we look at results obtained by the Random Forest (RF) Classification methods. All the features were considered to set a baseline for the accuracy to be expected by this method. The accuracy for this was obtained as 97.08%. Computing a confusion matrix for the same show that there still was 1 + 4 False Positives and False Negatives. This was the base accuracy expected from Random Forest and a further attempt to reduce the number of features was done without compromising the accuracy achieved with all of the features. Here the details that were found about the features during the data visualisation process proved to be very useful. To rank the top features sklearn package's SelectKBest function was used. It is a scoring function that it applied to a pair (x, y). The scoring function returns an array of scores and SelectKBest simply picks up the K best scores or highest scores for each feature of x, to yield y.

Running SelectKBest on all the features to retain the top 5 features brought up texture_mean, concanve_points_mean, concavity_se, symmetry_se, area_worst as the top 5 features. Using them re-running Random Forest gave us an accuracy of 97.07%. This step helped us reduce the feature by $1/6^{th}$ of the original set. But this method did see a slight step back in results by getting 2 + 3 False Positive and False Negatives. It is essential but to ensure that the feature reduction does not cause a drop in the accuracy. Therefore, the last optimization that was done for the Random Forest Approach was to test Recursive feature elimination (RFE) [1]. This is a good approach to not just find the most suitable features but also the most optimal number of features that ideally should be used. Using this the results showed us that the features number that was most optimal was 13. Once all the features other than the optimal features were removed and Random Forest only on those 13 were run then the output was 96.49%. This showed a slight reduction in the output accuracy generated.

The Classifier was also tested on various testing and training ratios to understand what is often the best ratio to be used Universally for Classification and Training. By changing the testing-training set we achieve the results as shown in the Figure 13.

Figure 13: Accuracy (%) vs. Testing-Training Ratio for Random Forest

The results obtained by using Random Forest has been 96.49%. The results here derived were computed after running the classifier multiple times. The results received with all features is still higher with 97.08% still one but pays in the price in number of features required to compute the same.

C. *Classifications using Logistic Regression*

The next method tested is by using Logistic Regression. The result obtained by this method was very varied initially. Using all features when the classification was executed the results were completely skewed, and a clear outcome was not obtained. Next, using the feature results obtained from running RF was considered. First the top 5 features were selected (results that were obtained by running SelectKBest). Using this the accuracy achieved was 96.5%. Again, to understand the optimal testing-training ratio the classifier was run on multiple to understand the best. The results of the tests are shown in Figure 14.

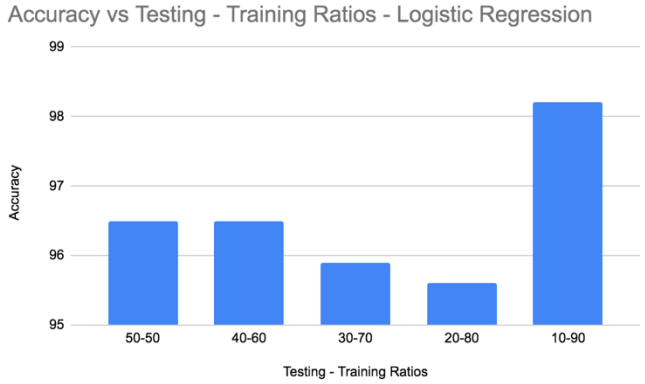

Figure 14: Accuracy (%) vs. Testing-Training Ratio for Logistic Regression

This shows a significant leap in the accuracy outcomes associated with the 10-90 testing ratio setting. Achieving a maximum accuracy of 98.2%. It is clearly significantly much higher and can be attributed to the low test ratio. It is unlikely to use such a drastically different result as the realistic outcome of the classifier compared to the previous ratio, and hence such results cannot be viable. It is safe to assume that the best achieved are the ones achieved using the 40-60 ratio which is 96.5%.

D. *Classifications using k-Nearest Neighbors*

Various parameters were tested for the K-Nearest Neighbor (k-NN) approach. Initially to set the baseline right for the accuracy to be obtained from k-NN, it was run using all of the features on the dataset. The 3 metrics tested are Euclidean, Manhattan and Minkowski. The results for the kind of distance metric did not majorly affect the computations here, proving that any of the distance methods can be used.

Given these results, the Minkowski distance metric was used. Hence so far, the maximum accuracy seen is 97.66%. To obtain these results for each method all values of k were tried to find the most optimal k value. K values were taken in range of 1 to 31, the total number of features being taken into consideration.

Another test was carried out by applying PCA before running k-NN to observe if it affected results.

Now to analyze how the number of components in PCA affects the outcome. Using 2, 5, 6 give the best results as shown in the following chart. Beyond 7 we can again see a decrease in the accuracy.

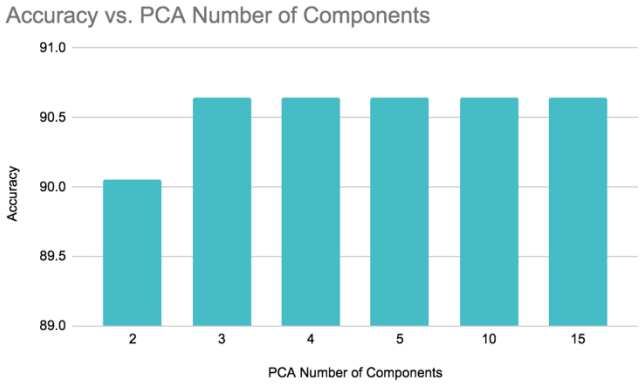

Figure 15: PCA Components vs Accuracy (%) for k-NN

We can observe here that beyond 3 components the results achieved by this methods have plateaued and offer no improvement. There is an overall dip in the accuracy. This proves that PCA should not be applied here.

Now, taking the all of the features and analyzing the output for all values of k is plotted in the graph displayed in Figure 17. From the figure we can see easily visualize the accuracies achieved for various values of k.

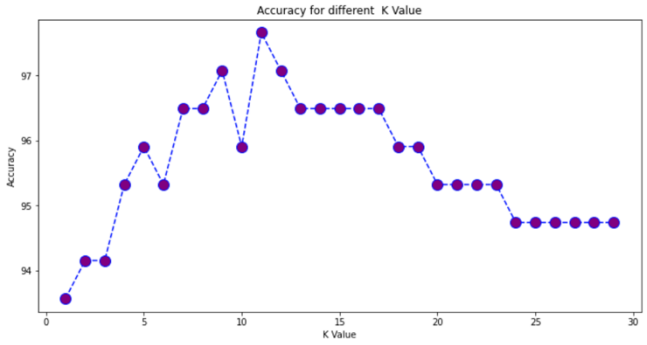

Figure 16: k-Values vs Accuracy (%)

Repeating the tests for all testing-training ratios the results displayed in Figure 18 were obtained.

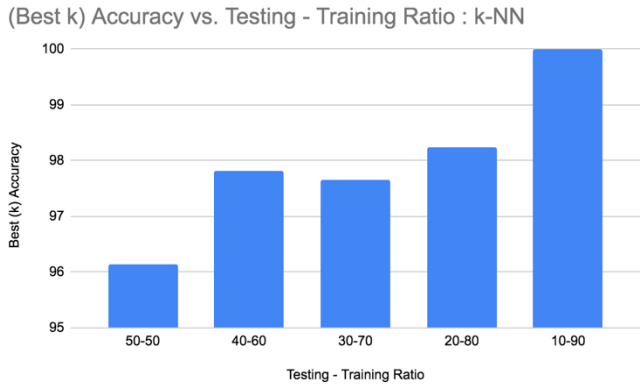

Figure 17: Accuracy (%) vs. Testing-Training Ratio for k-NN

The results are again very highly varied, and it is difficult to assess the best ratio to be used. Considering the patterns from the other classifiers, as well as making sure enough data is used for both training and testing, a ratio of 30% for testing was taken and an accuracy of 97.66% was achieved for k-NN. This is considered based on the results that show a drastic increase beyond this ratio going upto a 100%, which for the size of the dataset indicates there was not enough testing data, hence those results are not considered.

E. Classifications using Support Vector Machine Classification Technique

As the data that we have is already labelled, that is, we have the outcome as Malignant and Benign for each data entry, we can implement supervised learning methods. This opens up the usage of Support Vector Machines (SVM). Again as PCA was applied in the previous method, there too PCA was applied. Using the results that the best number of components found was 2, and taking that into consideration SVM generated an outcome of 95% in the initial trial. After running multiple tests for different test-train ratios we get that results displayed in Figure 19. From this we can see that using a testing training for this optimally would lie between 20-30% as going lower than that decreased the amount of data used for testing as well as the accuracy. Hence taking the testing ratio to be 30% for SVM we achieve an accuracy of 96.49%.

Figure 18: Accuracy (%) vs. Testing-Training Ratio for SVM

F. Classifications using Artificial Neural Networks

Having used various classifiers to get a given outcome, so far it is observed the accuracy has been around 96%. To see if this can be further increased Artificial Neural Networks are implemented. Initially the test was done using a Simple Neural Network with only a Single Layer. Tests were run using the Gradient Descent Optimizer (package available in Tensorflow), to minimize over cost. The cost in this regard was computed using Sigmoid Cross Entropy which weighs the probability error in discrete classification tasks in which each class is independent and not mutually exclusive. This is applicable here as the classification of Malignant and Benign are independent. Running this single layer perceptron Neural Network with varying number of steps gave us the results as displayed in the Table 1.

Table 1: SLP accuracy varying steps

Steps	Model Prediction	Test Prediction
5000	97.49%	95.32%
10000	98.24%	94.74%
15000	97.99%	93.57%
20000	97.99%	91.23%

Moving forward we only use the test prediction to made accurate guesses on the overall accuracy. The same tests are repeated while using the Adam Optimizer, which uses the Adam method for optimization. It works well for data which is sparse or noisy and also uses a larger effective step size. It computes the moving averages of the features. Using a popular step set of 10000 we achieved as accuracy of only 56.72%. Which is drastically lower than that achieved using Gradient Descent, establishing it Gradient Descent) as the better alternative. The next thing to be tried was various activation function to check the ideal one to be used moving forward along with the Gradient Descent Optimizer. The results are summarized in Table 2.

Table 2: Comparison of Accuracy (%) Activation functions

Activation Function	Model Prediction	Test Prediction
Sigmoid	98.24%	94.73%
ReLU	97.48%	90.05%
Hyperbolic Tan (tanh)	97.48%%	89.47%

The results from this tell us that the Sigmoid function performed the best compared to the other functions. Using a linear function would not be our choice for the following tests as well as the combination if a linear function with another layer applying a linear function would just be linear, possibly not yielding any better result at all even with the additional layer because of interference from the initial layers. With the hyperbolic function it is possible to hit values that cause very little change in the resulting values.
These were the results obtained using a simple Neural Network (NN) with no pre-processing done. Using PCA initially on the data and then running it through the same simple NN gives us changed results.

Table 3: SLP with PCA : Varying Number of Components vs Accuracy (%)

No. of Components	Model Prediction	Test Prediction
5	97.48%	97.66%
6	97.48%	96.49%
7	97.48%	97.07%
8	97.49%	96.49%
9	97.23%	96.49%
10	97.99%	96.49%

From the results displayed in Table 3 it is clear that best beyond 8 components the accuracy plateaued. The best possible accuracy achieved is 97.07% with 5 components. To further increase the possibility of getting a better result a more complex NN is implemented. Instead of a simple layer, now 2 hidden layers are considered and then the final output layer. The form of the neural network is as described in the Figure 20.

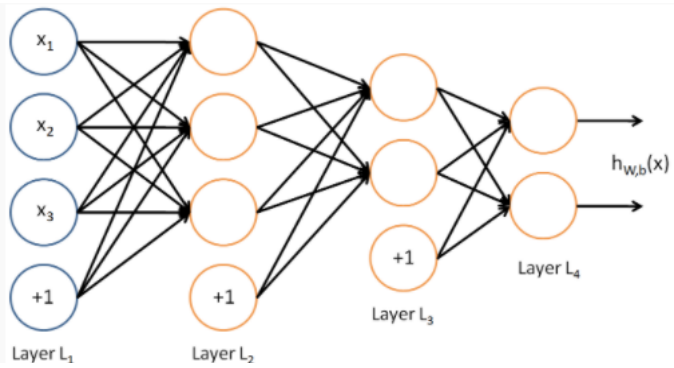

Figure 19: MLP Implementation Architecture [5]

Executing this kind of Multi – Layer – Perceptron (MLP) format and pre-processing using PCA again gave us a final outcome as shown in Table 4. For this implementation only, Gradient Descent was used as that proved to be the better optimization technique. The highest achieved accuracy from this method is using 5 components of PCA and with MLP, of 97.07%. This method reduced the number of components required without compromising the accuracy possible to be achieved.
Lastly adjusting the learning rate to achieve the highest possible accuracy gives us the results as shown in Table 5.

Table 4: MLP with PCA : Varying Number of Components vs Accuracy (%)

No. of Components	Model Prediction	Test Prediction
5	97.73%	97.07%
6	97.99%	96.49%
7	98.49%	95.90%
8	98.49%	95.32%
9	98.49%	95.90%
10	98.99%	95.32%

Table 5: Learning Rate vs Accuracy

Learning Rate	Model Prediction	Test Prediction
0.005	97.73%	97.07%
0.01	97.73%	97.07%
0.02	98.24%	96.49%
0.03	98.74%	95.90%

From this classification method the highest achieved accuracy achieved, is 97.07%.

V. FINAL OBSERVATIONS

After running the various classification techniques for the dataset, we can observe that for the given dataset, where the outcomes are binary, that is either Malignant or Benign only the best possible outcome was achieved using k-Nearest Neighbors (using all of the features), results in an accuracy of 97.66%. Principal Component Analysis with MLP and adjusting the learning rate to 0.01, using the Gradient Descent Optimizer gave the next best results of 97.07% along with Random Forest using all features by giving a result of 97.08%. These methods gave higher results than the other methods implemented. The highest accuracy after this was achieved using Logistic Regression after selecting the highest-ranking features. The results are summarized in Figure 21.

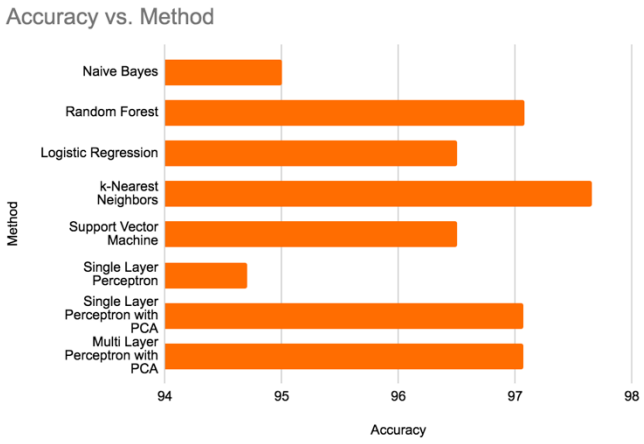

Figure 20: Summary of Results Achieved

Another conclusion that can be drawn is regarding the selection of the most “ideal” or optimal testing-training ratio. All of the classifiers the accuracy substantially increased when taking the ratio below 20%. This is possible for two

reasons, the higher training set, trains the model substantially better yielding better results, or the testing set is simple too small to make an accurate measure. It is highly unlikely that a simple increment in the training dataset can make such substantial increases in the accuracy, sometimes by very large margins, hence making it possible that the testing set is too small. From these experiments the optimal test ratio can be found to be in the range of 40-25% of the dataset, preferably closer to the median of the range.

This study of classifying also proved that all of the features collected in the dataset do not contribute to the final output and that out of the 30 features, the optimal number of features was around only 13. Reducing the noisy features would help generate more accurate results using only the most relevant features. These results could help focus more on data collection around these field in the future dropping some features like, 'smoothness_mean', 'compactness_mean', 'symmetry_mean', 'fractal_dimension_mean', 'radius_se', 'texture_se', 'perimeter_se', 'smoothness_se', 'compactness_se', 'concavity_se','concave_points_se','symmetry_se','fractal_dimension_se','smoothness_worst','compactness_worst', 'symmetry_worst', 'fractal_dimension_worst'.

The main difficulty observed in this problem statement was the collection of data. Medical data is not always widely available due to copyright reasons as well as the pre-processing required that makes such data usable by the scientific community. But in the recent years we are overcoming this as a community as a whole and moving to Open-Source availability of information. This will definitely see more scientists collaborating to produce much better medical technologies.

VI. Conclusion

After running various classifiers for mutually independent outcomes we can tell that trying to incorporate maximum number of features to the outcomes often skews the results due to certain features being loosely related to the result. In some cases, having a lot of outlier information, skewing the results and thereby decreasing the overall outcome's accuracy.

With the results obtained we can clearly see the advantage of using k-Nearest Neighbors for classification problems. With data that is fed into the algorithm for classification, there is no need for it to be linearly classifiable, yet the implementation is simple and hence has fast computation. The only time consuming task with using this algorithm proved to be finding the best k value possible. In this study all possible values of k, using the total number of features was tested, using a brute force approach. For future implementation, especially with datasets having much larger number of features this potentially might not be possible, needing an algorithm more performant approach.

While trying to consider multiple features using a Neural Network with multiple layers provided the best possible outcomes. Having hidden layers allows for individualization of the training data, by provided the necessary discrimination.

VII. Future Work

Many potential extensions for this comparative analysis can be done. Better classification models using the NN-MLP architecture can be implemented on various other datasets to build more information about the same. More training data can be added to achieve better outcomes. Another advantage of more data is that long term dependencies between features can be studied to provide better insights. Another following step could be to extend towards cancer research by understanding which features more likely affect a Benign or Malignant tumor, giving more data points to medical professionals. Light can be shed using this study on the highest features that lead to a certain cancer being Malignant or Benign aiding medical and bioinformatic research and boosting diagnostics, by eliminating human error.

VIII. Acknowledgement

I would like to thank Professor Dapeng Wu for guiding me and teaching important concepts in Pattern Recognition in an innovative and highly engaging manner. I was able to learn multiple concepts and theories over the semester and found the course highly exciting and informative.

References

[1] Genetic Analysis Workshop 20: envisioning the future of statistical genetics by exploring methods for epigenetic and pharmacogenomic data, n. d., https://bmcgenet.biomedcentral.com/articles/10.1186/s12863-018-0633-8

[2] K-NN Classification, https://www.saedsayad.com/k_nearest_neighbors.htm

[3] Breast Cancer Facts & Figures by the American Cancer Society, https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/breast-cancer-facts-and-figures/breast-cancer-facts-and-figures-2019-2020.pdf

[4] Chen, Huiling & Yang, Bo & Liu, Jie & Liu, Da-You. (2011). A support vector machine classifier with rough set-based feature selection for breast cancer diagnosis. Expert Systems with Applications. 38. 9014-9022. 10.1016/j.eswa.2011.01.120.

[5] Multi-Layer Neural Network, http://deeplearning.stanford.edu/tutorial/supervised/MultiLayerNeuralNetworks/

[6] Differences between Malignant and Benign Tumor, https://www.verywellhealth.com/what-does-malignant-and-benign-mean-514240

[7] Breast Cancer Wisconsin (Diagnostic) Data Set, https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

[8] Logistic Regression, https://saedsayad.com/logistic_regression.htm

[9] K-NN Classification, https://www.saedsayad.com/k_nearest_neighbors.htm