**I.**          **INTRODUCTION**

**1.1 Description of the project**

Breast cancer is the most commonly occurring cancer in women and the second most common cancer overall.(World Cancer Research Fund), what makes it a significant health problem for today’s society. According to clinical statistics, 1 out of 8 women is diagnosed with breast cancer during their lifetime. However, periodic clinical analysis examinations and self-examinations help in its early detection and thus it can facilitate timely clinical management of patients and help greatly increase the chances of survival. The method, employed worldwide for Breast Cancer detection consists of combination of 3 medical tests. After these, the clinicians flag the patients as having Breast Cancer if at least one of them indicates positive malignancy and as benign if all of them indicate benign. Therefore, the correct diagnosis of Breast Cancer and the classification of patients into malignant and benignant groups has become subject of multiple studies.

**1.2 Related work**

Major part of these studies use the Breast Cancer datasets from the University of California Irvine (UCI). Asri et al. (2016) investigate four different classifiers like Decision Tree, K-Nearest Neighbor (k-NN), NB, SVM and find that SVM gives the best classification accuracy of 97.13% and outperforms, therefore, all other algorithms. Borges (2015) achieved 97.80% by using the Bayesian Networks algorithm. Kharya et al., (2014) uses Naive Bayes (NB) algorithm for breast cancer detection and demonstrates the accuracy results of 93%. Nithya and Santhi (2014)  achieved 97.8% accuracy using an ensemble algorithm called multiboost Sequential Minimal Optimization. Maglogiannis et al. (2009) used SVM based classifiers and obtained the classification accuracy of 97%. Liu and Zheng (2006) obtained 92.90% accuracy, using filtered and supported sequential forward feature, which is based on support vector machine.

**II.**        **DATA DESCRIPTION**

**2.1**

We use the Breast Cancer Wisconsin (Diagnostic) Data Set, which is available from the online UCI database. This data set was created by Dr. William H. Wolberg, physician at the University Of Wisconsin Hospital at Madison, Wisconsin,USA. To create the dataset Dr. Wolberg used fluid samples, taken from patients with solid breast masses and an easy-to-use graphical computer program called Xcyt[1], which is capable of performing the analysis of cytological features based on a digital scan. It contains 569 instances and 32 attributes. It includes 30 features and 1 explanatory variable. The output variable corresponds to the diagnosis of breast tissue, which is either “M” for malignant or “B” for benignant.   Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nucleus present in the image. All the inputs are continuous and real-valued and computed for each cell nucleus. The main features are:

1. Radius mean: mean of distances from center to points on the perimeter

2. Texture mean: standard deviation of gray-scale values

3. Perimeter mean: mean size of the core tumor

4. Area mean

5. Smoothness mean: mean of local variation in radius lengths

6. Compactness mean: mean of perimeter^2 / area - 1.0

7. Concavity mean: mean of severity of concave portions of the contour

8. Concave points mean:mean for number of concave portions of the contour

9. Symmetry mean

10.  Fractal dimension mean

         The mean, standard error and “worst” or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

 (See if we can make a box plot for the features)

Approach –

Given the number of training model and techniques at our disposal, we would like to understand which algorithms can be best suited according to our classification needs.

In this project we take gradual steps in building our understanding towards different classification model which encompasses the following –

i)           A comparison of efficacy of various Machine Learning Methods

ii)          Developing understanding on feature selection

iii)         Finally, optimizing the hyperparameters to tune the algorithm to provide best possible classification

Th motivation is not just to be able to train a Model to provide better accuracy but we would also like to optimize the number of features that are required. Given the complex nature of Medical Sciences we require best possible prediction with minimum number of features and selected.

**Disclaimer:** The tools and techniques used in this project are being extensively taught in the lecture course. So our focus majorly will be on building the concept to achieve better classification results. For this we assume the reader has equal and/or more knowledge of the concepts from the lecture. Hence, we would only dive deeper into mathematical details whenever it's explicitly required. Though we would certainly mention the key differences in various models as we proceed further

**MODEL SELECTION:**

We have divided the Model Selection into 3 sub sections -

Part 1: Preliminary Comparison - This is the first step our analysis where we train various Machine Learning Models to make an initial comparison on the performance of various models. We use all the information available to us - essentially we use all the features to train the algorithms. This gives us a starting point into the efficacy of various models.

Part 2: Subset Selection: The most fundamental expectation from an algorithm is to provide efficient results with the minimum number of required features. In this section we explore three basic approaches to have reduced dimensionality in order to

**Different Models Used**

**LDA and QDA**

'linear' (default) — Estimate one covariance matrix for all classes.

'quadratic' — Estimate one covariance matrix for each class.

 Discriminant analysis and nearest neighbor algorithms do not analyze data that contains both numeric and categorical variables. But in our case since we dont have categorical features we go ahead with LDA and QDA

**Gaussian Naive Bayesian:**

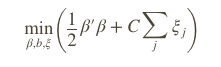
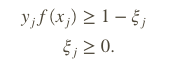
The fitcdiscr function has other two other types, 'DiagLinear' and 'DiagQuadratic'. They are similar to 'linear' and 'quadratic', but with diagonal covariance matrix estimates. These diagonal choices are specific examples of a naive Bayes classifier, because they assume the variables are conditionally independent given the class label.

**Decision Tree**

**Another classification algorithm is based on a decision tree. A decision tree is a set of simple rules, such as "if the sepal length is less than 5.45, classify the specimen as setosa." Decision trees are also nonparametric because they do not require any assumptions about the distribution of the variables in each class.**

**Support Vector Machines**

In this section we discuss the methodology we used with respect to our SVM analysis. We have not assumed linear separability in our dataset and hence we train our Model for Soft Margin SVM. The optimization problem is -

such that 

In this case our eta variable or the slacked variable ensures Soft Margin SVM and its zero for the case of Hard Margin. C is the penalty parameter and we will discuss the optimization of this hyper parameter in subsequent section.

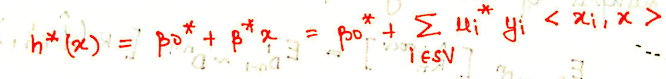
Just to provide a quick context - we know from theory that the Dual optimization problem derived for Hard and Soft Margin SVM is similar, the only difference is the constraints.

**How to classify ?**

The decision for any given x is constructed using



which essentially translates to finding the inner product  -



Now to produce better result we transform our feature space to a higher dimension Hilbert Space. Now instead of working the above solution with the inner product of xi we calculate the inner product of  <*φ*(*x*1),*φ*(*x*2)>. And we obtain this inner product of  <*φ*(*x*1),*φ*(*x*2)> using Kernel Trick

**Advantage**

The beauty of the techniques lies in obtaining a non linear decision boundary while still working with linear classification.

**Different SVM Methods Used**

We train our data for three different SVMs. Since we have 30 features its not possible to visualize the data and hence comment if the linear SVM is better. So we would like to explore further with other Kernel functions - Quadratic and Gaussian SVM

Model Selection -

Filtering

<https://de.mathworks.com/help/stats/examples/selecting-features-for-classifying-high-dimensional-data.html>

Filters are usually used as a pre-processing step since they are simple and fast. A widely-used filter method for bioinformatics data is to apply a univariate criterion separately on each feature, assuming that there is no interaction between features.

For example, we might apply the *t*-test on each feature and compare *p*-value (or the absolute values of *t*-statistics) for each feature as a **measure of how effective it is at separating groups.**

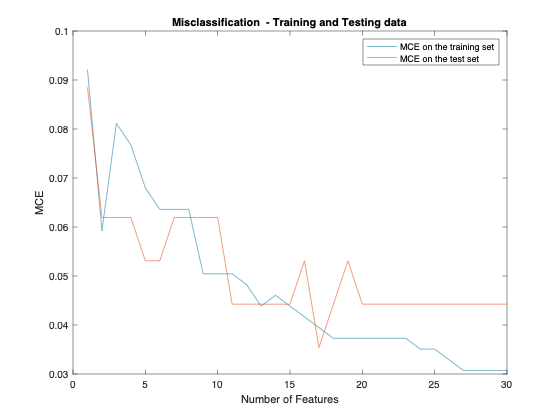
[h,p] = ttest2(x,y,'Vartype','unequal')

returns a test decision for the null hypothesis that the data in vectors x and y comes from independent random samples from normal distributions with equal means without assuming that the populations also have equal variances.

28    23    8      21    3      1      27    7      24    4      6      26    11    22    13    14    2      25    18    5        29    9      30    16    17    20    15    10    12    19

This ranking gives us first step in exploring feature selection. Using the t-test we got a preliminary ranking of features according to their importance in terms of discrimination power.

We then go ahead and try to try to train a basic Quadratic Disc Model just to observe how this feature ranking performs. In the next segment we start with the so called best feature, as obtained using t-test and we keep on adding the next subsequent important feature in each step. We train our algorithm with one extra feature each time and then we plot the Misclassification error for training and testing data.

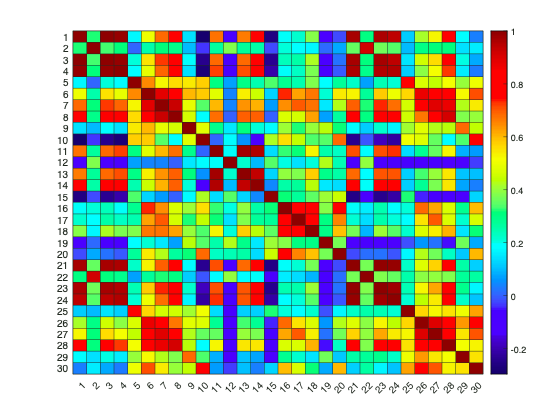


The figure gives a few interesting observations –

1. We see a gradual decrease in the training error as we increase the number of observations (except for second onbservation). We would also expect a drop in the training error as we increase the number of features as this is not a cross validation error.

2. The drop in Misclassification for second observation (i.e feature 23) hints that 23rd feature may be an important feature in our classification criteria. Or maybe our third observation which is 3rd feature might ne giving us wrong classification thereby increasing our error. We will further explore on this part in the report.

3. Lastly we do observe a drop in error with our testing data initially and then it doesn’t change much, which may be likely that the additional features do not play significant role in classification further.



**Drawback:**

A drawback with this approach may be that it does not account for correlation within the features

As we can observe their are features that are correlated with values above 85%. We will deal with this issue in coming section on Principal Component Analysis

**Forward Selection**

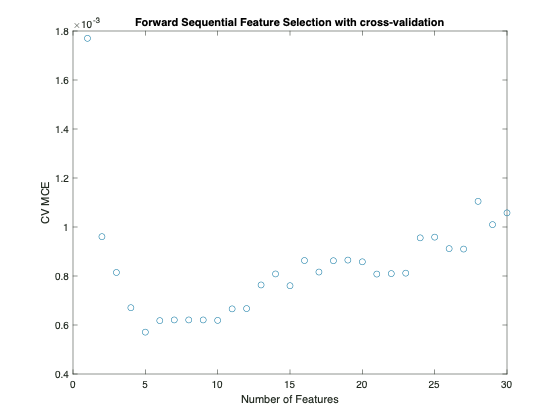
Till now we observed how our pre-liminary exploratory analysis gave initial inferences feature ranking in accordance with the Misclassification Error.

Now we instead of fixing which feature enters first in our analysis, as we did earlier based on p-value ranking, this time around we allow the algorithm to select best feature for itself in each round and calculate the corresponding error.

We proceed with Forward feature selection approach in which the algorithm chooses the feature corresponding to lowest error and in the subsequent round we keep on adding features to our already accumulated features till now.

**Cross-validation approach**

To ensure that our error calculations are robust we use a 10-Fold cross validation approach to obtain the corresponding error while training our algorithm



**Key Inference**

We observe that the algorithm keeps on selecting the best subsequent feature to keep on lowering the error. However, once top 5 features are added, the next subsequent addition to the features actually gives a model with higher error of decreased precision.

So we infer that top 5 features selected using forward selection provide us with relevant features to predict our model.

**NOTE**

Since the forward selection algorithm is Greedy in the sense that it tries to achieve the best result with limited number of iterations. We can not surely say that Forward Selection is the Universally best method to give us the least error model. There are essentially 2K-1+1 (including only intercept model) possible and its probable that due to greedy setup an omitted feature combination might still provide with a better model.

**NOTE**

Also we need to note that the Y axis is not exactly the error term. We have used Matlab function ‘sequentialfs’, which in turn uses cross validation approach and tries to minimize the criterion function. In our case we have used loss from classification and hence the criteria is to reduce the classification loss in each step. However, the ‘sequentialfs’ function sums up all the criterion ‘scalar’ value and divides it by the number of test observations. Hence the y-axis indirectly represents a comparison of classification error only that it is divided by number of test data by the function.

But for our purpose since we only require the ranking of the features we can consider these values as an indirect classification error.

**Feature Selection - Principal Component Analysis**

**Hyperparameter Optimization**

Once we have discussed the methods on subset selection, we proceed further to optimize the hyper parameters in our trained Models. We will allow for optimization of hyper parameters based on overall training data that we have, i.e. considering all the features. Due to computational issues we are only considering a few of the training algorithms in this section.

**Approach**

Different machine learning algorithms consider different methods to classify data and in this section we will briefly describe the hyperparameters which we intend to optimize.

1. **Polynomial SVM**

Our very first comparison earlier considered Linear and Quadratic Support Vector Machines. In this section we take a combined approach and put them under the polynomial SVM case which essentially uses the Polynomial Kernel- *G*(*x*1,*x*2) = (1 + *x*1′*x*2)^p.

We use Bayesian Optimization form the Built-in Matlab function to optimize the Box Constraint level or the soft-margin penalty known as C in the primal equations.

1. **Gaussian SVM**

Our next step in optimizing the hyper parameters is to optimize using kernel function - Gaussian type. The Gaussian Kernel used is - *G*(*x*1,*x*2) = exp(–∥*x*1–*x*2)∥2).

1. **Classification Tree**

The Box Constraint parameter is the soft-margin penalty known as C in the primal equations.

(*Mention stuff and explain a little bit of Soft Margin SVM)*

*Link* [*https://de.mathworks.com/help/stats/choose-a-classifier.html#buwh8ek*](https://de.mathworks.com/help/stats/choose-a-classifier.html#buwh8ek)

An alternative way to manage support vectors is to reduce their numbers during training by specifying a larger box constraint, such as 100. Though SVM models that use fewer support vectors are more desirable and consume less memory, increasing the value of the box constraint tends to increase the training time.

<https://de.mathworks.com/help/stats/support-vector-machines-for-binary-classification.html#bsr5o1q>

References

University of California Irvine. UCI Machine Learning Repository; 2019. Available from

Nithya R, Santhi B. A Data Mining Techniques for Diagnosis of Breast Cancer Disease. World Applied Sci J 2014:18-23.

Liu, Y. and Y.F. Zheng, 2006. FS-SFS: A novel feature selection method for support vector machines. Pattern Recognition, 39(7): 1333-1345.

Maglogiannis, I., E. Zafiropoulos and I. Anagnostopoulos, 2009. An intelligent system for automated breast cancer diagnosis and prognosis using SVM based classifiers. Applied Intelligence, 30(1): 24-36

Borges, L.R. Analysis of the Wisconsin Breast Cancer Dataset and Machine Learning for Breast Cancer Detection. Group 1989, 1, 369.

Kharya S, Agrawal S, Soni S (2014).Naive Bayes classifiers: A probabilistic detection model for breast cancer. Int J Comput Appl, 92, 26-31.

[1] The program uses a curve-fitting algorithm, to compute ten features from each one of the cells in the sample, then it calculates the mean value, extreme value and standard error of each feature for the image, returning a 30 real-valuated vector.