Spring 5-22-2020

Classification of Prostate Cancer Patients into Indolent and Aggressive Using Machine Learning

Yashwanth Karthik Kumar Mamidi
University of New Orleans, ymamidi@uno.edu

Classification of Prostate Cancer Patients into Indolent and Aggressive Using Machine Learning

A Thesis

Submitted to the Graduate Faculty of the
University of New Orleans
in partial fulfillment of the
requirements for the degree of

Master of Science
in
Computer Science

by

Yashwanth Karthik Kumar Mamidi

B.Tech. Gitam University – Hyderabad, 2018

May, 2020

Acknowledgments

First of all, I would like to express my sincere gratitude to my supervisors Dr. Md Tamjidul Hoque and Dr. Chindo Hicks, for providing their invaluable guidance, support, and suggestions required for the project.

I would like to thank Dr. Minhaz Zibran and Dr. Christopher Summa, for their kind consent to be a committee member for my thesis defense.

Next, I would like to thank my brother and lab partner Tarun Karthik Kumar Mamidi, for his assistance in providing me an overview of the project. I appreciate his guidance, helping nature, and inspiring criticism to contribute equally to this work. I would also like to thank my family and friends for their encouragement and support, which helped me in completion of this project.

I would also express my gratitude towards the University of New Orleans and LSU Health Science Center for providing me a creative space for research.

Table of Contents

List of Figures

List of Tables

Abstract

Prostate cancer (PCa) is the second most common cancer in men in the US. Many Prostate cancers are Indolent and don't result in cancer mortality, even without treatment. However, a significant proportion of patients with Prostate cancer have aggressive tumors that progress rapidly to metastatic disease and are often dangerous. Currently, treatment decisions for PCa patients are guided by various stratification algorithms. Among these parameters, the most important predictor of PCa mortality is the Gleason Grade (ranges from 6 to 10). Although current risk stratification tools are moderately effective, limitation remains in their ability to distinguish truly Indolent from aggressive and potentially lethal disease. Here we propose the use of Machine Learning (ML) for the classification of PC patients as having either indolent or aggressive using transcriptome data. We hypothesize that genomic alterations could lead to measurable changes distinguishing indolent from aggressive tumors. We also trained a Stacking-based model with a different set of combinations of classifiers. The highest overall accuracy of our stacking model (all samples with Gleason Grade: 6, 7, 8, 9, and 10) is 95.758% and (samples with Gleason Grade: 6, 8, 9, and 10) is 97.19%.

KEYWORDS: Machine Learning, Stacking, Prostate Cancer, Gleason Grade.

Chapter 1 – Introduction

Prostate cancer (PCa) is the most common solid tumor and the second most common cause of cancer death in the United States [1]. To date, treatment decisions for PCa patients are guided by various risk stratification algorithms [2]. These stratification algorithms are used for identifying and predicting the patients, who are at high risk or likely to be at high risk with the disease. Among the parameters used, the most potent predictor of PCa mortality is the Gleason grade (GG) [3, 4]. The GG ranges from 6 to 10. The majority of PCa present GG 6. These cancers are associated with very low cancer-specific mortality rates, even in the absence of therapy. Intermediated grade PCa presents GG 7. These cancers present a much more variable clinical course. Localized high grade (aggressive) with lethal potential PCa presents GG: 8 to 10. These tumors are aggressive, progress rapidly to metastatic disease, and are often lethal. Although current stratification protocols are moderately effective, significant challenges remain classifying PCas into Indolent and Aggressive. A key knowledge gap and critical unmet medical need are distinguishing patients with truly indolent tumors from those with aggressive tumors.

PCa screening using the prostate-specific antigen (PSA) has led to the earlier detection of PCa with fewer men today presenting with metastatic disease[5]. However, although PSA has led to a reduction in mortality rate, it has also resulted in unintended consequences. The unintended consequences include over-diagnosis, which leads to overtreatment of patients indolent PCa, and under-treatment of patients with aggressive disease. Concerns about PSA-based screening led to the issuing of a D grade recommendation of its use by the US Preventive Services Task Force in 2012 [6]. Crucially, a review for the U.S. Preventive Services Task Force concluded that PSA-based

screening results, either small or no reduction in prostate cancer-specific mortality [7]. It is associated with harms related to subsequent treatments and evaluation - some of them may be unnecessary. These concerns have heightened the need for the development of novel risk stratification algorithms to identify patients at high risk of developing aggressive tumors, which could be prioritized for treatment, and discovery of molecular markers separating the truly indolent disease from aggressive disease.

Here we propose the use of machine learning (ML) for classification of PC patients into two groups, those with genuinely indolent tumors and those with aggressive tumors using transcriptome data. Statistics does simpler things, and when coming to a complex environment, it would be hard to predict. Implementing ML can help in predicting things more accurately and come up with better results. Our working hypothesis is that genomic alterations in patients diagnosed with indolent and aggressive could lead to measurable changes distinguishing the two patient groups, and that application of ML to genomics data would accurately distinguish the two patient groups. We addressed this hypothesis using transcriptome data on patients diagnosed with indolent and aggressive PCa from The Cancer Genome Atlas (TCGA).

Chapter 2 – Literature Review

2.1 Background and Related Works

Prostate cancer is characterized by malignant tumors found within the prostate gland in menage 65 and older. Currently, it is diagnosed with a blood test called Prostate Specific Antigen (PSA) test. Various attempts were made to classify cancer-based tissue samples using microarray, clinical, imaging, and RNA sequencing data. A new approach is developed to improve accuracy when using microarray data for classification [8]. Some of the recent studies attempted to diagnose prostate cancer with machine learning utilized microarray datasets. Few of them conducted using various methods and were tested on different datasets [9-11]. They aim to predict if cancer is metastasizing or not, and the results of all microarray datasets are significant.

The TCGA database is already used for classifying different types of cancers, and the data contained goes beyond RNA sequencing data in the TCGA database. Few published studies used breast cancer datasets for cancer classification [12, 13]. The challenges associated with datasets from the TCGA database are class imbalanced and are high dimensionality. If the dataset is high dimensional, the model cannot separate the classes accurately, and the result obtained will be very poor. Moreover, If the dataset is a class imbalance, the number of features will be much more than the number of samples, and the model becomes unstable and cause overfitting problem. Few studies faced the same problem using TCGA datasets [12, 13].

Lei Yang *et al.* [14] used Random walk with restart algorithm (RWRA) and Graph-regularized Nonnegative Matrix Factorization (GNMF) methods for molecular classification of prostate adenocarcinoma by the integrated somatic mutation profiles and molecular network. They

analyze somatic point mutations in exome sequences from TCGA-prostate samples and obtained with better results.

In one of the recent study [12], they used Stacked denoising Autoencoder (SDAE), PCA, KPCA, and differentially expressed gene methods to reduce the dimensionality. They also tried different methods like Artificial Neural network (ANN), Support-Vector Machine (SVM), Support-Vector Machine (SVM) with linear kernel, and Support-Vector Machine with Radial basis function kernel (SVM-RBF). The highest accuracy was obtained by SVM-RBF using the SDAE method for dimensionality reduction, and the highest sensitivity is achieved by the ANN model, followed by the SDAE method. The highest specificity and precision are obtained by the SVM-RBF model. Glocuk *et al.* [13] aimed to increase accuracy by performing different dimensionality methods like PCA, KPCA, and NMF. They tried implementing ladder network and found that it is outperforming SDAE and AVM models.

Takumi *et al.* [15] tried machine learning to diagnose prostate cancer using clinical data. They implemented an Artificial Neural Network (ANN) with the data and found that although their model performed well, improvements need to be made before being suitable for clinical applications.

A recent study [16] implemented the SMOTE technique to increase the number of samples in the data set to deal with imbalanced class. Using Smote, they created synthetic observations and equalized the class distribution. They applied the Recursive Feature Elimination (RFE) algorithm to reduce the number of features to identify the tumor. Later, they performed a logistic

regression model using 5-fold cross-validation to minimize the false positive rate and improved the accuracy compared to previous machine learning attempts.

Jaideep *et al.* [17] aimed to classify prostate cancer using a protease activity nanosensor library and tried to identify aggressive disease on a different dataset. They implemented a bottom-up approach to design nanosensors to classify and detect prostate cancer. To identify proteolytic enzymes in human prostate cancer, they used Transcriptomic and proteomic analysis. They also tried measuring the activity by building a library of nanosensors. Moreover, they demonstrated that these nanosensors could classify aggressive tumors and outperformed a serum maker in mouse models. This library can be used at the screening test to identify patients with higher-risk tumors.

Lemana *et al.* [18] developed an Artificial Neural Network (ANN) to classify normal and prostate cancer patients. They obtained the dataset from the research done by Zhou *et al*. [19]. They used Prostate-Specific Antigen (PSA) levels and Mitochondrial DNA copy number (mtDNA) samples. They aimed to classify samples with 175 normals and 177 tumors (according to biopsy results). The best performance is obtained with two-layer feedforward ANN with a log-sigmoid transfer function. The log-sigmoid (log-sig) transfer function is also used in a multilayer network, which uses the backpropagation algorithm. Moreover, they applied 10-fold cross-validation and resulted in sensitivity as 100%, specificity as 98.8%, and overall accuracy as 99.4%. They used a different dataset, which is obtained from the research done by Zhou *et al*. [19] and tried to separate healthy samples (normals) with diseased samples, but we are trying to distinguish samples with two diseases (Indolent and Aggressive).

2.2 Review of Machine Learning Methods

In this section, we describe the usage of machine learning methods and their underlying principles. We also explained the reason in the introduction section, why we choose the machine learning methods.

2.2.1 Support Vector Machine (SVM)

A Support Vector Machine (SVM) [20] is a machine learning classifier, which is defined by a separating hyperplane. Support Vector Machine algorithm finds a hyperplane in an N-dimensional space that classifies each data point (where N is the number of features).

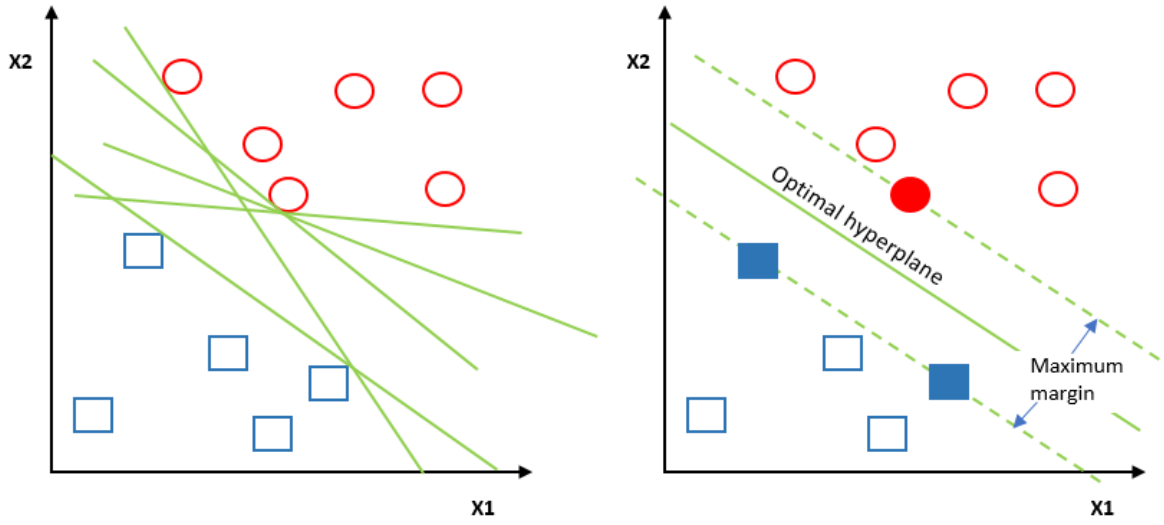


Figure 2.1: A two-class classification problem is shown in the above figure. The left side figure shows cases where data points may be separated from many different decision boundaries. The right side figure represents the optimal hyperplane that has the highest margin and is considered the decision boundary.

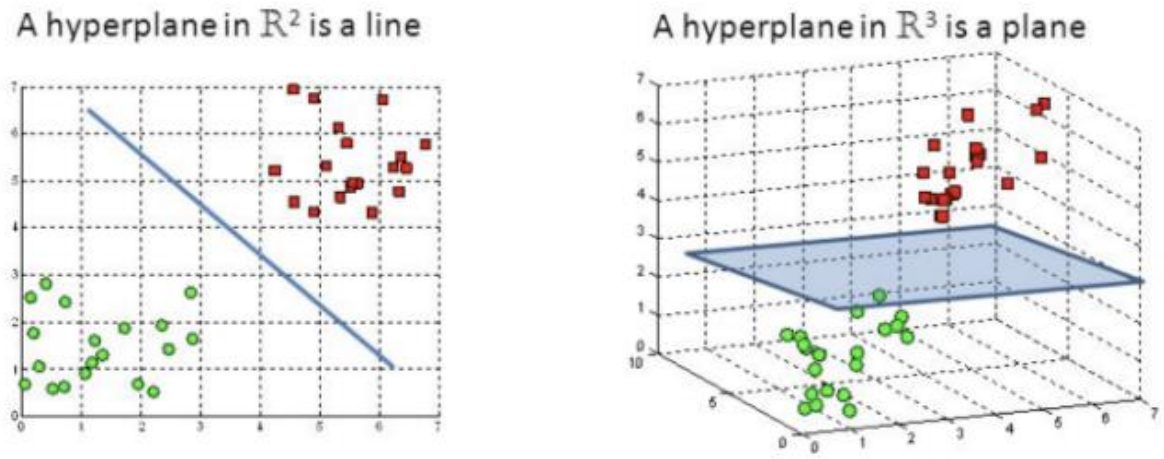


Figure 2.2: The Left-side graph represents hyperplane separating two classes in 2-dimension as a line, and right-side graphs show hyperplane separating two classes in 3-dimension.

Hyperplanes help in classifying data points and depends upon the number of features. If the number of features in a dataset is 2, then the hyperplane is just a line. If the number of features in a dataset is 3, then the hyperplane is a plane. If the number of features is greater than 3, then it would be difficult to imagine a hyperplane.

2.2.2 Logistic Regression (LogReg)

Logistic Regression [21] is a technique for analyzing data that determines the dependent output (outcome) when there are one or more independent variables. In several cases, the outcome variable (dependent) is a dichotomous variable, in which there are only two possible outcomes. The goal is to find the best fitting model to describe the relationship between the dependent variable and the set of independent variables. Logistic sigmoid (log-sig) function is used to return a probability value by transforming the output, which can be mapped to discrete classes. Regularization techniques are used to avoid overfitting (any modification made to a learning algorithm is intended to reduce the generalization error).

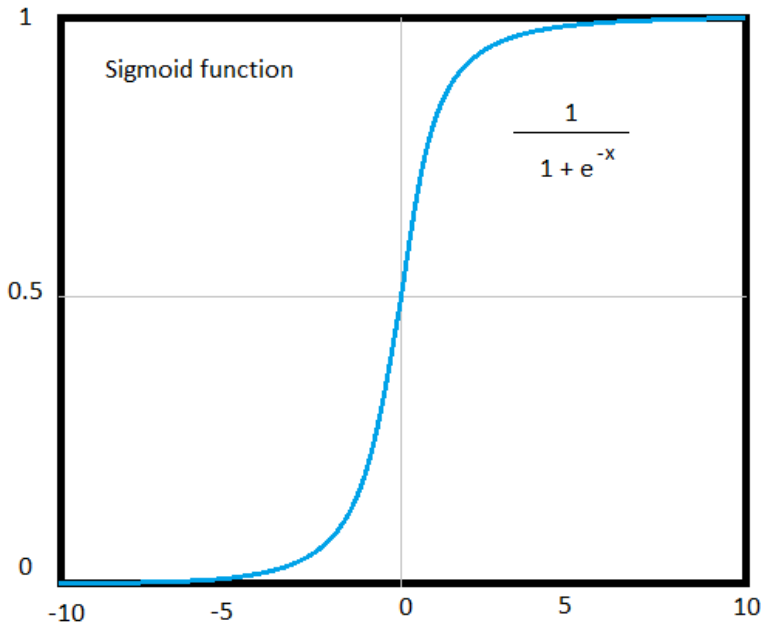


Figure 2.3: The sigmoid function takes a real value and maps it to the range [0, 1]. The decision function is used to obtain the probability of class.

2.2.3 Random Decision Forest (RDF)

Random decision Forest [22] is a supervised machine learning algorithm which randomly creates and merges more than one decision tree into a forest. During training time, Random Decision Forest (RDF) algorithm operates by constructing a multitude of decision trees and outputting the class that is Classification or mean prediction (regression) of individual trees. It adds additional randomness to the model growing the trees. The best feature is searched among a random subset of features, instead of searching for the most crucial feature while splitting a node. Random decision forests correct habit of overfitting to their training data-set. The RDF operates by constructing a multitude of decision trees on various subsamples of the dataset and results in a mean prediction of decision trees to improve accuracy and avoid over-fitting.

2.2.4 Extra Tree Classifier (ETC)

The Extra Tree [23] method is also known as extremely randomized trees. The main objective of an Extra Tree classifier is to randomize the input features of a tree, where the large proportion of the variance of the induced tree depends on the choice of optimal cut-point. It constructs randomized decision trees from the original learning samples and uses the above-average decision to improve accuracy and avoid over-fitting. The method selects a cut point at random and drops the idea of using bootstrap copies of the training sample. Cut-point randomization often reduces the variance, when the bootstrapping idea is drooped and can also lead to an advantage in terms of bias. This method has yielded state-of-the-art results in high dimensional complex problems.

2.2.5 Gradient Boosting Classifier (GBC)

Gradient boosting classifier [24] is a machine learning technique used for classification and regression problems. It builds a model in a forward stage-wise fashion like other boosting methods. It allows for optimizing arbitrary differentiable loss functions. It involves three elements: (a) a loss function to be optimized, (b) a weak learner to make predictions, and (c) an additive model to add weak learners to minimize the loss function. The main objective of the Gradient boosting classifier is to minimize the loss of the model by adding weak learners in a stage-wise fashion using a similar procedure of Gradient descent. While adding a new weak learner, the existing weak learners in the model remain unchanged. In order to correct or improve the final output, the output of a new learner is added to the existing sequence of learners.

2.2.6 K Nearest Neighbors (KNN)

K nearest neighbor [25] is an algorithm that classifies new cases based on a similarity measure of all stored available instances. It has been used as a non-parametric technique in statistical estimation and pattern recognition. A case is being assigned to the common class among the K nearest neighbors, which is measured by a distance function and is also classified by a majority vote of its neighbors. If k=3, then the class is assigned to a class of its three nearest neighbors shown in Figure 2.4.

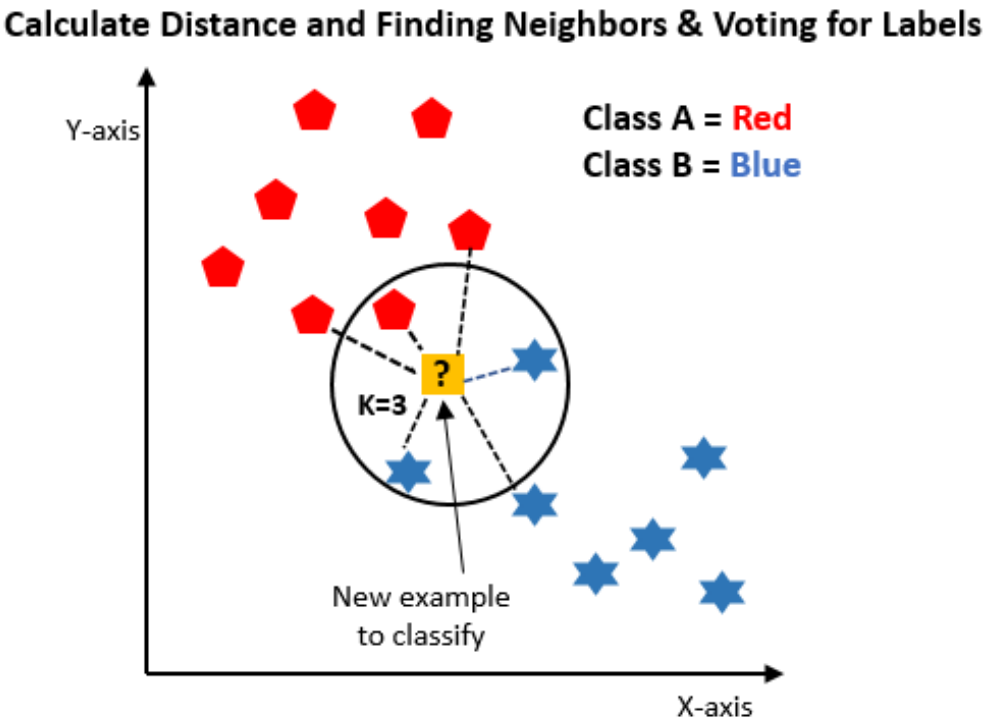


Figure 2.4: The figure above shows the Calculation of distance and finding neighbors and voting for the KNN method.

2.2.7 eXtreme Gradient Boosting (XGB)

The implementation of eXtreme Gradient Boosting [26] offers several advanced features for model tuning, algorithm enhancement, and computing environments. It can perform in three different forms of gradient boosting (Gradient Boosting (GB), Stochastic Gradient Boosting (GB), and Regularized Gradient Boosting (GB)). It is strong enough to support fine-tuning and addition of regularization parameters. It uses the regularized model formalization to avoid overfitting and results in better performance. Moreover, XGB trains faster.

2.2.8 MultiClassClassifier

MultiClassClassifier in WEKA is used for handling multi-class datasets with 2-class distribution classifiers. It is also capable of applying error-correcting output codes for increased accuracy. If the weights are not uniform, the base classifier cannot handle instance weights. So the data will be resampled with a replacement before being passed to the base classifier. It extends RandomizableSingleClassifierEnhancer and implements OptionHandler and weightedInstancesHandler.

2.2.9 Logistic Model Trees (LMT)

The logistic model tree (LMT) [27] is a classification model with a logistic regression function at the leaves. It is made up of an inner or non-terminal node along with a set of terminal nodes. It predicts a continuous numeric value for an instance that is defined over a fixed set of attributes. It constructs a piecewise linear approximation to the target function. LMT consists of a tree with a linear regression function at leaves. For instance, it is obtained by sorting it down to a leaf and also by using the prediction of the linear model associated with that leaf. It doesn't

incorporate all the attributes present in the data in order to avoid building overly complex models.

Chapter 3 – Experimental Materials and Methods

3.1 Sources of Transcriptome and Clinical Data Sets

We used publicly available gene expression and clinical data on indolent and aggressive PCa from the TCGA. The data were downloaded from the Genomic Data Commons [28], data portal using the data transfer tool. Because the same TCGA barcode structure was used for both clinical data and transcriptome data, we used the barcodes structure to integrate patient-based clinical data with sample-based genomics data. The total data set included N = 547 samples distributed as follows: N = 45 samples on indolent (GG=6), 246 samples with intermediate (GG=7), 204 of aggressive with lethal potential and 52 control samples. Gene expression data used in this thesis were derived from the same patient population. After annotating gene expression data with clinical information, we used the American Urological association classification protocol to verify and validate the classification of tumors according to GG because GG =7 follows a variable clinical course. We used the protocol to assign the tumors to either indolent or aggressive consistent with the guidelines. The tumor samples were either classified as 3 + 4 (primary + secondary), or 4 + 3 (primary and secondary) grade. The samples with GG: 3 + 4 grades were assigned to a group of patients with Gleason Grade 6 (Indolent PCa). The samples with GG: 4 + 3 grades were assigned to a group of patients with Gleason Grade 8 to 10 (Aggressive PCa) [29].

We performed data quality control and processing steps on gene expression data containing 60,483 probes across 547 samples. We implemented CPM (counts per million) filter (>0) in R to remove the rows with missing data, such that each row had at least ≥ 30% data. After

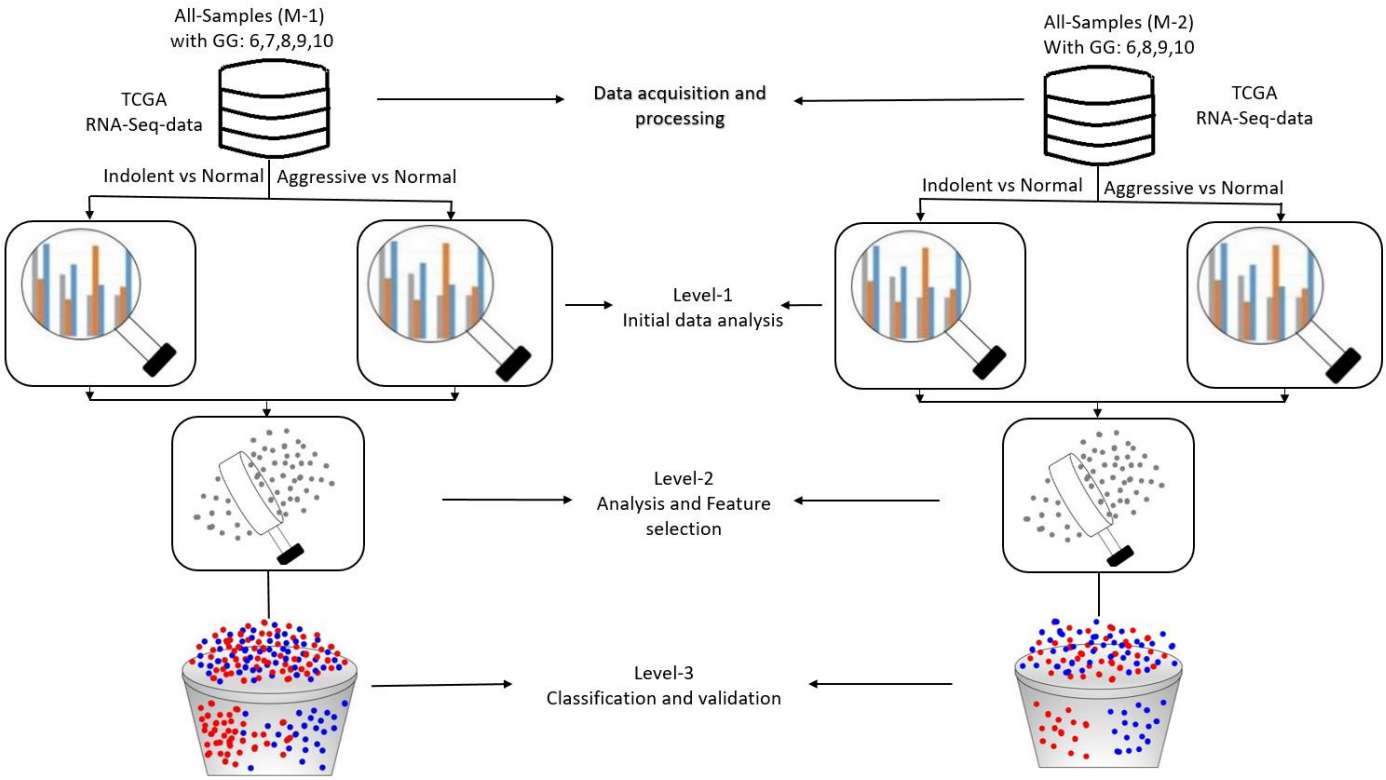


Figure 3.1: Flowchart depicting project design and execution workflow in this project. Only the genes significantly differentially expressed between tumors and controls discovered in level 1 analysis were considered in the level 2 analysis. COAD: colon adenocarcinoma; DE: differential expression; TF: tumor-free; TP: tumor presenting.

filtering the data, we obtain a new dataset with 34,956 probes across 547 samples. We corrected the data for the library sizes for all the samples in with gene expression data. The resulting data set we normalized using CPM function to get log2 counts per million and checked for distribution properties.

3.2 Data Processing and Analysis for Gene Selection

Using the limma and edgeR packages in R 3.8.0 [30], we processed the data and performed quality control by removing probes with low or zero expression values. The remaining data set was normalized using quantile normalization. Data normalization was performed using TMM. Composition biases are eliminated between libraries and generated a set of normalization factors (the product of the library sizes and factors defines the effective library size) using TMM normalization. TMM normalization scale relative to one sample and normalization factors multiple to unity across all libraries. Below Figure 3.4 shows the biased and unbiased MD plots side by side for the same sample (acb3e352-b255-4c41-b90f-5e5ed2273b06) before and after TMM normalization. Implemented in R before performing statistical tests. The processed normalized data contained 34,956 probes.

3.2.1 Level-1 Analysis

Using normalized data in R [30], we performed level 1 analysis comparing gene expression levels between tumor samples and controls for indolent and aggressive PCa separately. We used this baseline analysis to discover a signature of genes significantly ($p < 0.05$) associated with each disease state. We used the false discovery rate (FDR) procedure to correct for multiple hypothesis testing. The probes were ranked on *p*-values and –log fold change (-log FC). Differentially expressed significant probes between tumors and controls were considered to be associated with PCa. This initial level 1 analysis yielded 18,215 significantly ($p < 0.05$), and 21,042 significantly ($p < 0.05$) differentially expressed probes associated with Indolent and with aggression for model 1. For model 2, This initial level 1 analysis yielded 15,105 significantly ($p < 0.05$), and 20,712

significantly (p < 0.05) differentially expressed probes associated with Indolent and with aggressive. Additionally, an analysis comparing Gleason grade 7 yielded 15,105 significantly (p<0.05), and 20,712 significantly (p < 0.05) differentially expressed probes associated with Indolent and aggressive for model 1.

The significant probes were then matched to the corresponding gene symbols and screened for duplicates. This analysis resulted in 3513 genes used in the downstream analysis in level 2 and classification. In addition to feature selection, multiple data visualization tools were also used to identify the most significant subset of probes for further analysis, including volcano plot, principal component analysis, and hierarchical clustering to heat-maps. Hierarchical clustering was performed only on the most highly significant genes to assess similarity in patterns of gene expression among the genes associated with the disease. For hierarchical clustering, we used the Pearson correlation as the measure of the distance between pairs of genes, and complete linkage as the clustering method. Hierarchical clustering was performed using Morpheus (Versatile matrix visualization and analysis software).

We test for differentially expressed genes using our normalized data, and there are many packages to analyze RNA-Seq data. Limma package offers the voom function, which transforms the read counts into logCPMs while considering the mean-variance relationship in the data. We created a design matrix for the groups and made the column names of the design matrix a bit nicer. Here, the decision matrix tells us which samples correspond to each group. Now, we perform voom transformation using our decision matrix, and it will adjust the library sizes using the norm.factors already calculated and generate a plot of mean-variance trend (shown in Figure 3.6 below). We can tell, if there are any genes that look really variable in our data and if we have

filtered low counts adequately using this plot. Below, Figure 3.3 shows the boxplot for the normalized data with expression log-transform values to compare to before normalization.

Now we used limma to test for differential expression using voom transformed data. First of all, we fit a linear model for each gene in limma using the lmFit function. lmFit needs the design matrixand the Voom object that are already specified, which is stored within the voom object. Since we are interested in differences between groups, we need to specify which comparisons we want to test by specifying the comparison of interest using makeContrasts function. Here we get the statistics and estimated parameters of our comparison by using contrasts.fit function in limma. The final step is performing empirical Bayes shrinkage on the variance and estimates moderated t-statistics and the associated *p*-values by calling eBayes function and to generate a quick summary of DE genes for the contrast we used limma decideTests function. We used the volcano plot (shown in Figure 3.7 below) using the functions in limma for plotting the data with fit.cont as input.

3.2.2 Level-2 Analysis

We performed level 2 analysis on both the significant genes associated with the indolent and with the aggressive disease using gene expression data. We also compared gene expression levels between patients with Gleason grade 6 versus patients with Gleason grade 8-10. Moreover, we compared patients with Gleason grade 6 (3+4) versus patients with Gleason score 8-10 (4+3). Patients presenting with Gleason grade 3+4 versus patients presenting with Gleason grade 4 + 3 were also compared. Indolent and Aggressive patients (Ind Vs. Agg) to identify the features or genes to be used in classification algorithms. False discovery rate (FDR) procedure is

used to correct for multiple hypothesis testing. The genes were ranked on *p*-values and –log fold change (-log FC). Genes significantly differentially expressed between disease states were using the classification algorithms.

3.2.3 Level-3 Analysis

Application and evaluation of classification algorithms were involved using selected probes or features in this analysis using different cut-offs as determine by the *p*-values and logFC from the 2074 genes identified in the analysis. To test and validate the classification algorithms, few features were selected by performing feature selection at different threshold levels using the Genetic Algorithm. According to Machine Learning literature, five classifiers were selected with different fundamental approaches: Logistic Model Tree (LMT), MultiClassClassifier, SGD, SMO, SimpleLogistic. We also performed the stacking technique with few other classifiers with different fundamental principles:

(a) Support vector machine (SVM).

(b) Logistic Regression (LogReg).

(c) Random Decision Forest (RDF).

(d) Extra Tree Classifier (ETC).

(e) Gradient Boosting Classifier (GBC).

(f) K nearest neighbor (KNN).

(g) eXtreme Gradient Boosting (XGBoost).

To address the deficiency, we standardized the data and later processed it using a class-balancing algorithm due to the unstable design of the project. This algorithm is applied to each classifier as

well after the poor initial performance. Five subsets of 2074 significant genes were taken based on log-fold-change cutoffs of 0.5, 0.7, 1, 1.5, and 2. Here, we used 10-fold cross-validation technique on all mentioned subsets to prevent overfitting, with metrics averaged over all 10 folds and tested on each classifier. Weka 3.8.2 software [31] and the Genetic Algorithm are used to perform all classification and evaluation.

In Figure 3.2, all the library sizes of samples in TCGA data are expressed using a barplot to see whether there are any major discrepancies between samples. It shows that the data quality is not good and is not normally distributed. To examine the distributions of raw counts, we need to log the counts. Here, we used box plots to check the distribution of the read counts on the log2 scale. Figure 3.3 represents the boxplots of logCPM (log counts per million) before normalization.

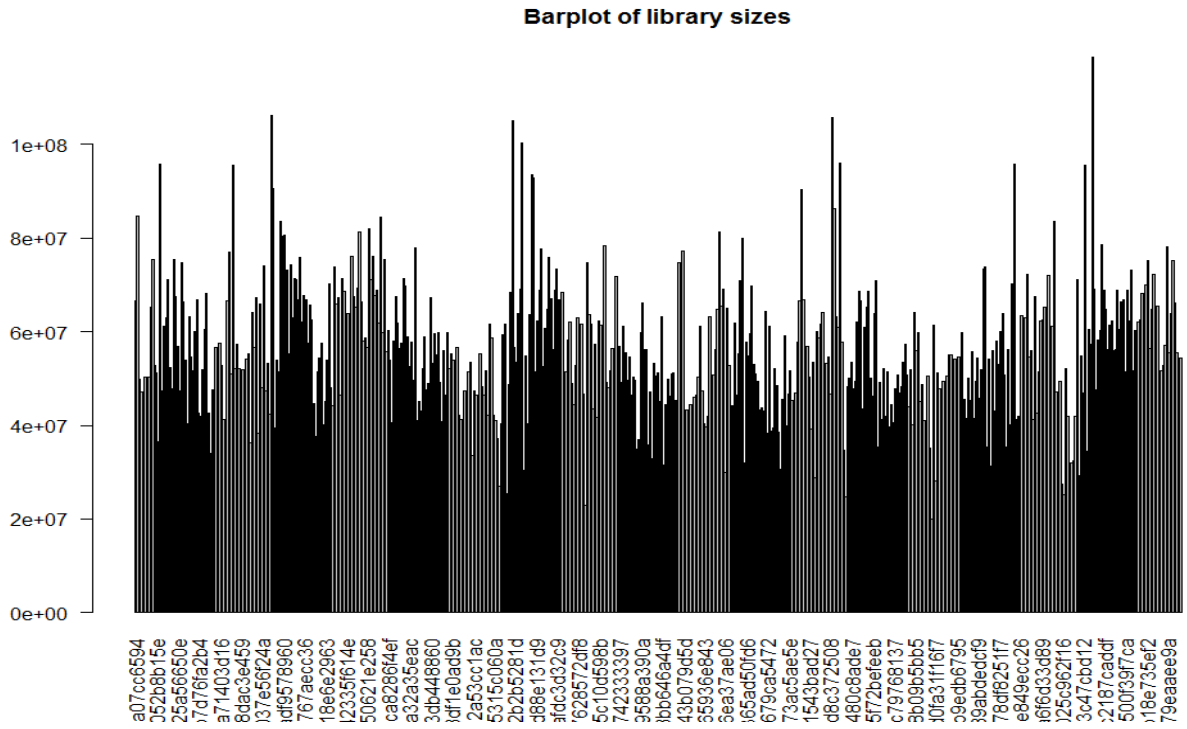


Figure 3.2: Library sizes of all samples expressed using a barplot constitutes the data quality and unnormalized library sizes.

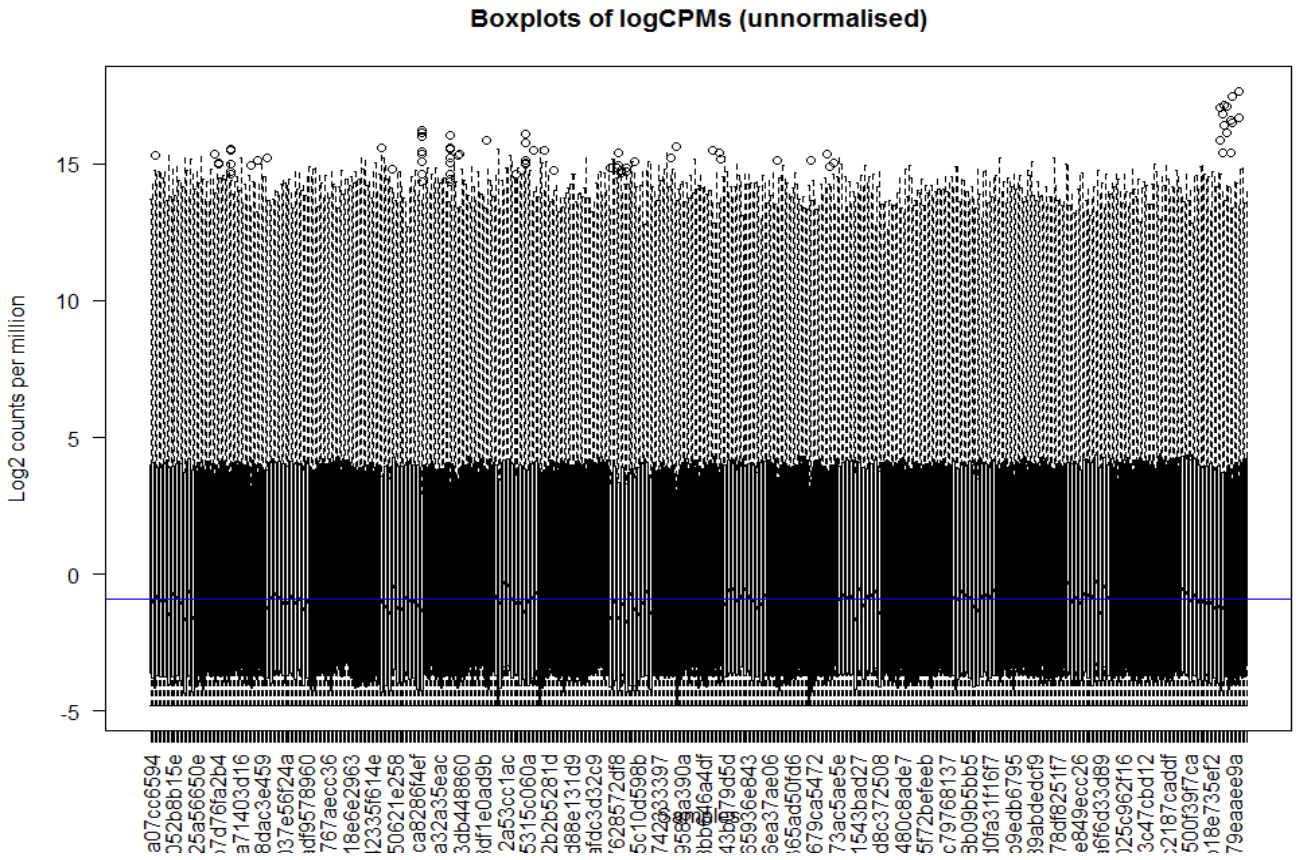


Figure 3.3: Figure checks the distribution of the read counts on the log2 scale of logCPM (log counts per million) before normalization.

3.2.4 Normalization for Composite Bias

We used TMM normalization to eliminate composition biases between libraries and generated a set of normalization factors (the product of the library sizes and factors defines the effective library size) [32]. TMM normalization scale relative to one sample and normalization factors multiple to unity across all libraries. Below Figure 3.4 shows the biased and unbiased MD plots side by side for the same sample (acb3e352-b255-4c41-b90f-5e5ed2273b06) before and after TMM normalization.

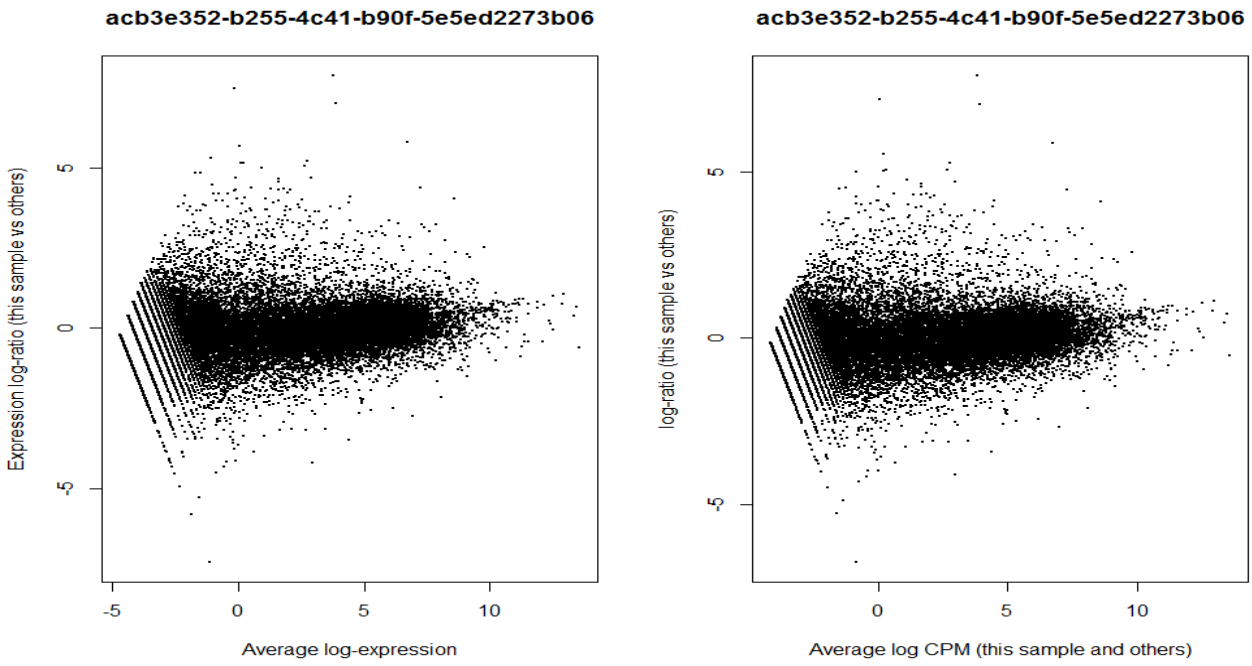


Figure 3.4: Biases and unbiased MD plots side by side for the same sample.

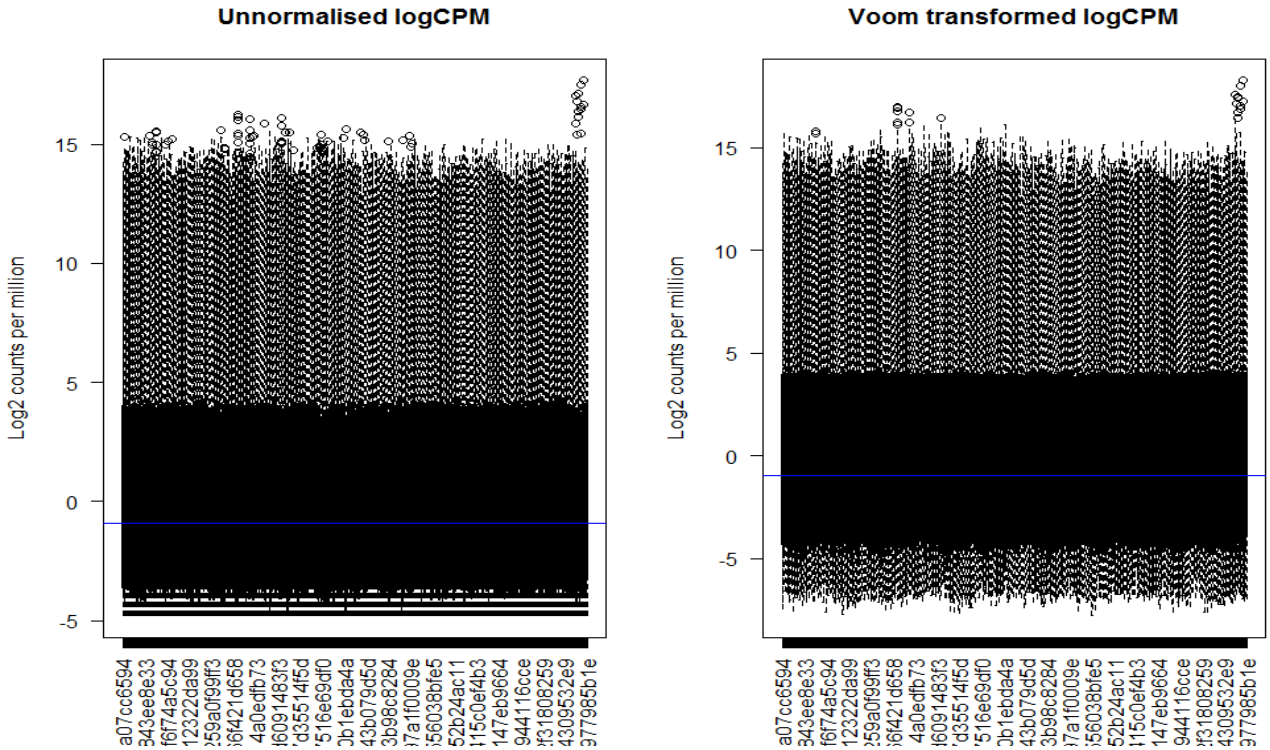


Figure 3.5: Figure represents the comparison of the data set before and after normalization of logCPM log counts per million.

3.2.5 Differential Expression with limma-voom

We test for differentially expressed genes using our normalized data, and there are many packages to analyze RNA-Seq data. Limma package offers the voom function, which transforms the read counts into logCPMs while considering the mean-variance relationship in the data. We created a design matrix for the groups and made the column names of the design matrix a bit nicer. Here decision matrix tells us which samples correspond to each group. Now, we perform voom transformation using our decision matrix, and it will adjust the library sizes using the norm.factors already calculated and generate a plot of mean-variance trend (Figure 3.6). We can also say that if there are any genes in our normalized dataset that look really variable using this plot and if we have filtered low counts fairly.

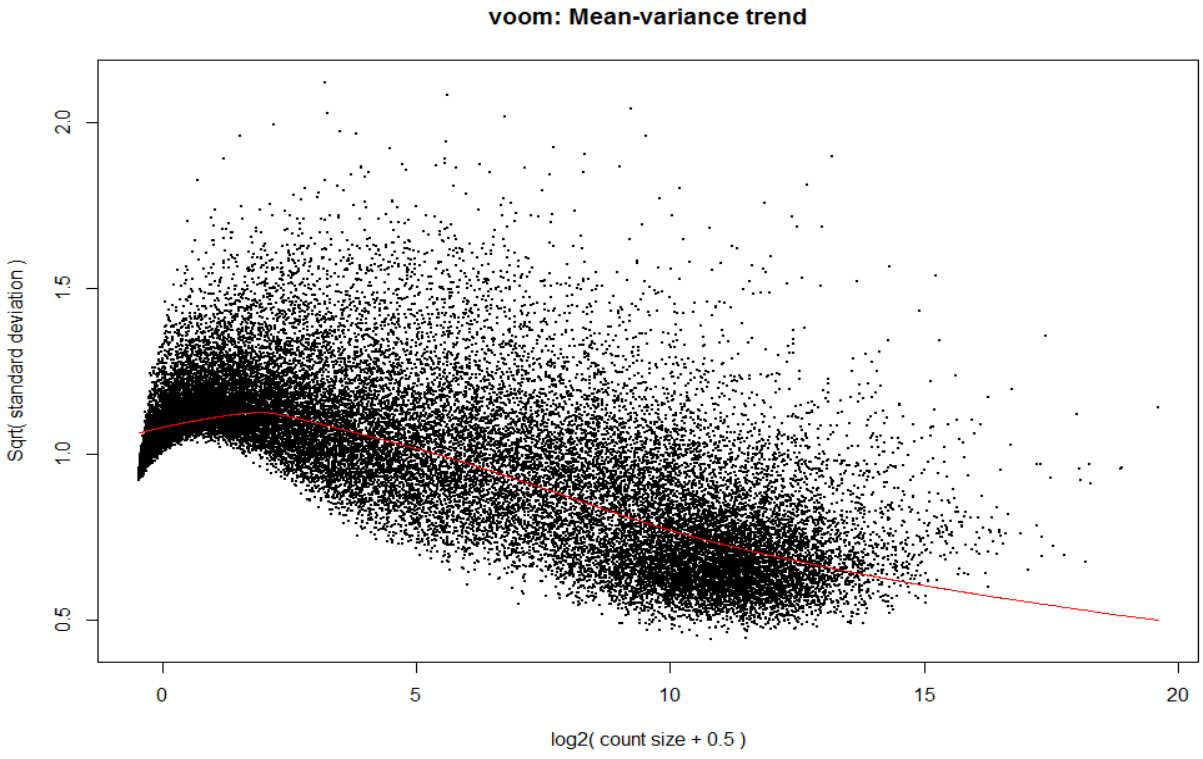


Figure 3.6: Figure showing the voom transformation using a decision matrix and mean-variance trend.

3.2.6 Testing for Differential Expression

Now we used limma to test for differential expression using voom transformed data. First of all, we fit a linear model for each gene in limma using the lmFit function. lmFit needs the design matrix and the Vvoom object that are specified previously, which is stored within the voom object. Since we are interested in differences between groups, we need to specify which comparisons we want to test by specifying the comparison of interest using makeContrasts function. Here we get the statistics and estimated parameters of our comparison by using contrasts.fit function in limma. The final step is performing empirical Bayes shrinkage on the variance and estimates moderated t-statistics and the associated *p*-values by calling eBayes function and to generate a quick summary of DE genes for the contrast we used limma decide Tests function. We used the volcano plot (Figure 3.7) using the functions in limma for plotting the data with fit.cont as input.

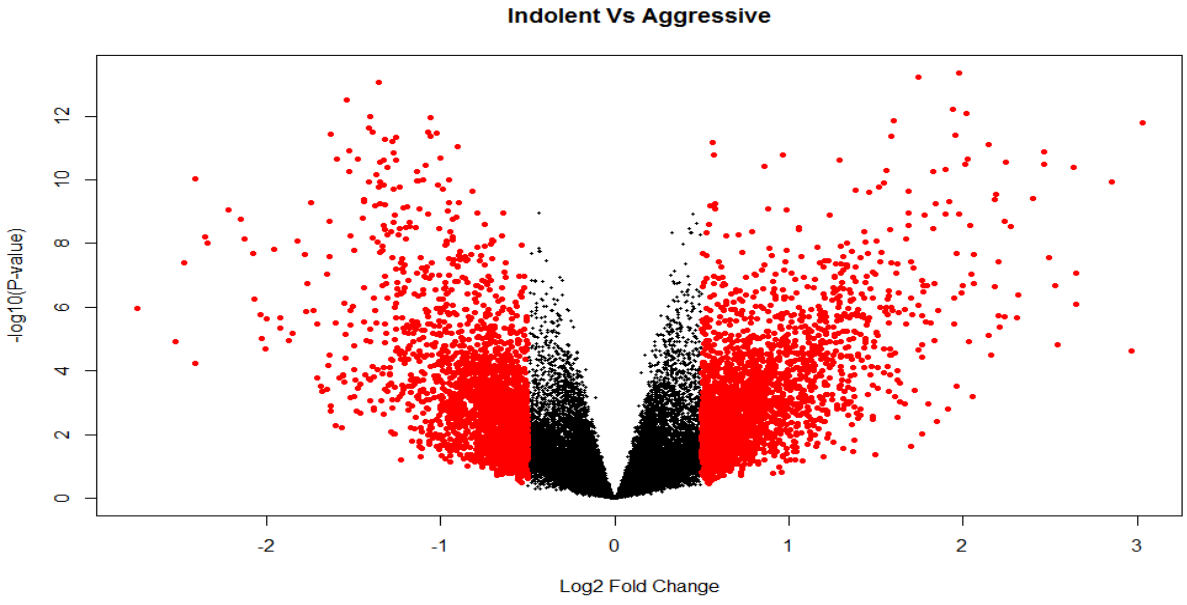


Figure 3.7: Figure showing that we used the threshold values to obtain the probes associated with 2 different diseases (Indolent and Aggressive).

3.2.7 Multidimensional scaling plots (MDS)

Multidimensional scaling plots (MDS) is a visualization of a principal components analysis, which determines the sources of variation in the given data. We used MDS-plots after analyzing our RNS-Seq data set. To make the plot more informative, we colored the samples (Aggressive: blue, Indolent: red) according to the grouping information and plotted them using points. Leading log fold change is used to calculate the distance between each pair of samples in the MDS plot, defined as the root-mean-square of the largest 500 log2-fold changes between that pair of samples.

3.2.8 Hierarchical clustering with heatmaps

Hierarchical clustering is an alternative for examining the relationships between samples. Heatmaps are a nice visualization to examine hierarchical clustering, and it is done using heatmap.2 function from the gplots package. It will calculate a matrix of Euclidean distances from the logCPM (logcounts objects) for the top 500 most variable genes from our normalized dataset. The top 500 most variable genes across samples are shown in the heatmap.

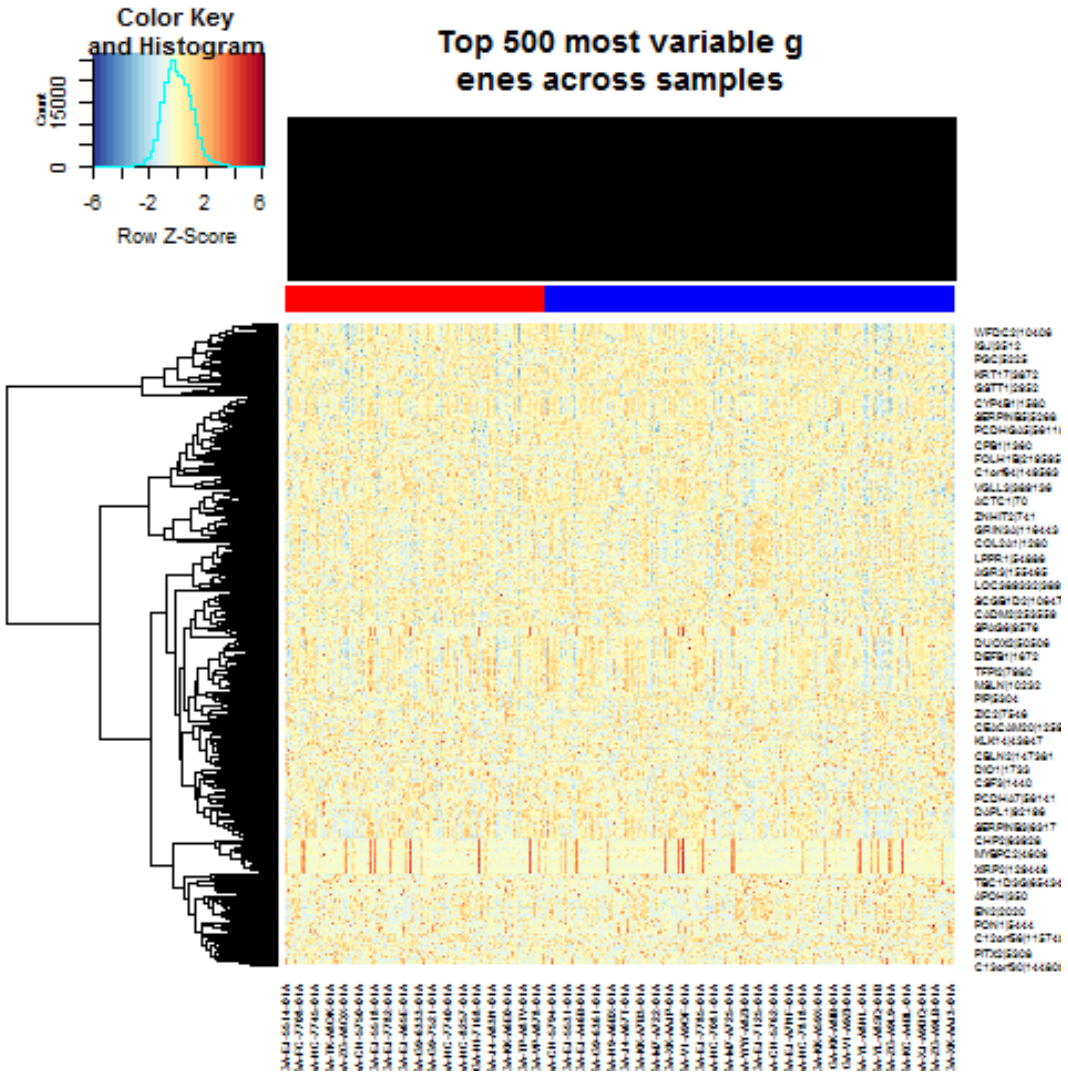


Figure 3.8: Figure representing a matrix of Euclidean distances from the logCPM (logcounts objects) for the 500 most variable genes.

3.3 Application of Machine Learning

We perform differential expression (D.E.) using both data sets (voom-normalized and clinical) and by comparing Indolent VS Normal and Aggressive Vs. Normal, we generate two probe sets with values of logFC, *p*-values, adjacent *p*-values, etc.

Now we filter the probes with adjacent *p*-values <0.05 in both the data sets (Indolent Vs. Normal, Aggressive Vs. Normal) and merged these two datasets by eliminating the common probes. We filter the latest normalized data with all the unique probes we obtained from the two datasets (Indolent Vs. Normal, Aggressive Vs. Normal). Applied Differential Expression (D.E) on this data and generated a Probe set (Indolent Vs. Aggressive) by comparing Indolent Vs. Aggressive. Later on, filtered the probe set with adjacent *p*-values <0.05 along with different log fold change values (0.5, 0.7, 1, 1.5, and 2).

We further filtered the generated latest normalized data with the probes obtained from the probe set (Indolent Vs. Aggressive) and removed the normal samples. Now, we apply 10-fold cross-validation on the filtered normalized data and find the error rate. As the error rate is high, we can conclude that the data set containing samples with Gleason grade 7 are more misclassified.

Now, we remove the samples which have Gleason grade 7 from the normalized dataset and perform differential expression (D.E) using both data sets (Normalized (without GG: 7) and clinical), and by comparing Indolent VS Normal and Aggressive Vs. Normal, we generate two probe sets with values of logFC, *p*-values, adjacent *p*-values, etc.

We filter the probes with adjacent *p*-values <0.05 in both the data sets (Indolent Vs. Normal, Aggressive Vs. Normal) and merged these two datasets by deleting the common probes. We filter the latest normalized data (without GS: 7) with all the unique probes, we obtained from the two datasets (Indolent Vs. Normal, Aggressive Vs. Normal). Applied Differential Expression (D.E) on this data and generated a probe set (Indolent Vs. Aggressive) by comparing Indolent Vs.

Aggressive. Later on, we filtered the probe set with adjacent *p*-values <0.05 along with different log fold change values (0.5, 0.7, 1, 1.5, and 2).

We again filtered the generated latest normalized data (without GG: 7) with the probes obtained from the probe set (Indolent Vs. Aggressive) and removed the normal samples. Now, we apply 10-fold cross-validation on the filtered normalized data and find the error rate. As the error rate is low, we can say that almost all samples in the normalized data set (without GG: 7) are classified correctly.

Here, we have classified above that the normalized data set with the samples (without GG: 7) correctly. So, we take the normalized dataset (without GG: 7) for training and the normalized data set with samples (only GG: 7) for testing. Now, we apply 10-fold cross-validation on these data sets and find out the samples which are misclassified.

3.4 Correlation between ML and GG = validation

3.4.1 Model-1:

We applied a machine learning approach on all the probes with all the samples (495) with Gleason Grade (6, 7, 8, 9, 10) for all log-fold change values (0.5, 0.7, 1, 1.5, 2). We observed that the correctly classified instances in the normalized data set represented in Figure 5.1 is approximately 80%. The remaining 20% is mostly caused because of the misclassification in the samples (246) with Gleason Grade: 7 (3+4 and 4+3). We conclude that few samples with Gleason Grade: 6, 8, 9, and 10 are also misclassified. The below table represents the number of probes for each log-fold-change value.

Figure 5.4 in the result section shows that most of the samples with Gleason Grade: 7 are misclassified after applying the machine learning approach using different classifiers.

3.4.2 Model-2:

Here, we assume the samples (246) with Gleason Grade: 7 (3+4 and 4+3) are misclassified and removed these samples from the normalized data and apply machine learning on the remaining data set with samples (249) using the 5 classifiers mentioned above.

Figure 5.12: in the result section shows that most of the samples (249) with Gleason Grade (6, 8, 9, and 10) are classified with almost 90% correctly classified instances. We can also say that there are few incorrectly misclassified instances in these samples, too (Gleason Grade: 6, 8, 9, and 10).

So, we used the samples (249) with Gleason Grade (6, 8, 9, and 10) to find the misclassified samples (246) with Gleason Grade: 7 by using machine learning.

We applied machine learning on all the samples after classifying the misclassified instances with Gleason Grade: 7. Figure 4.1 shows the number of misclassified cases in the complete data set.

Chapter 4 – Performance Evaluation

In the section, We plot a graph with misclassified instances in Figure 4.1. We performed machine learning approach on the normalized data-set using five different classifiers based on their principles and observed that the misclassified instances in samples with GG: 7 are more using MultiClassClassifier classifier and also found that the total number of misclassified instances in the whole data-set is more using Logistic Model Tree (LMT) followed by SimpleLogistic classifiers. The below-mentioned Table-1 and Table-2 contain the log-fold change values and the number of probes associated with the disease for both models (Model-1 and Model-2). We also created a performance evaluation matrics with their names and definitions mentioned in Table-3 below.

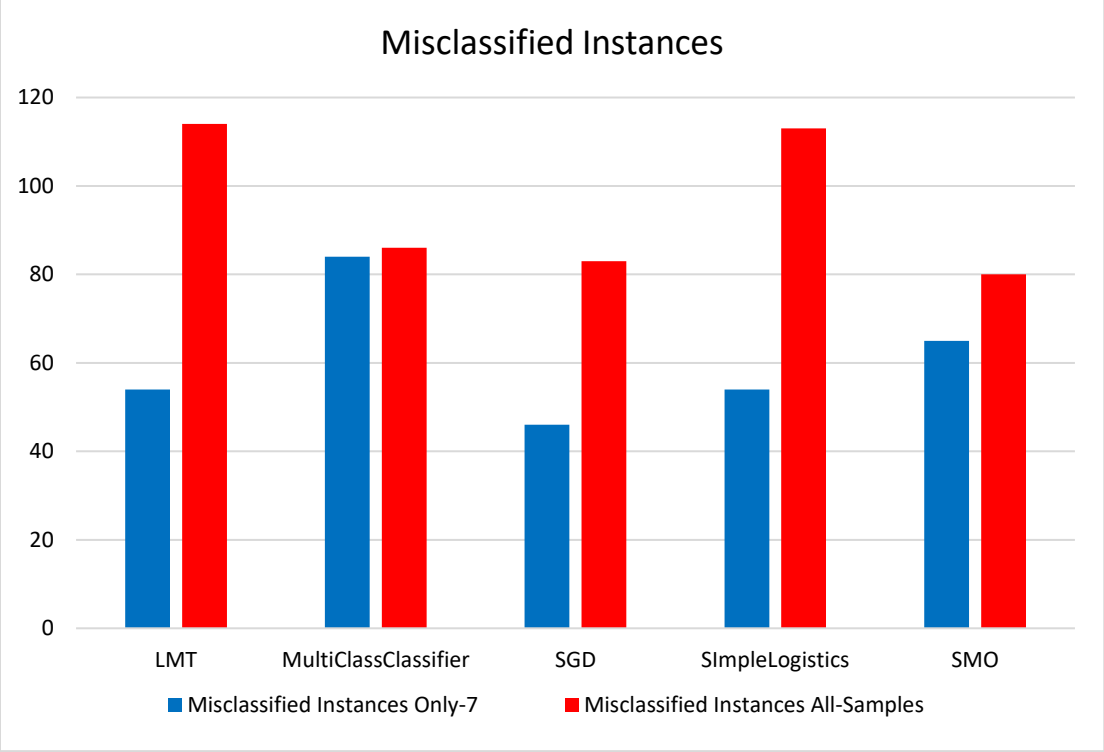


Figure 4.1: Figure representing the misclassified instances in samples with both datasets (Samples with GG: 7 and Samples with GG: 6, 7, 8, 9, 10).

Figure 5.6 in the result section represents the normalized data set after classifying all the misclassified instances (495 samples) using a machine learning approach with different threshold values (0.5, 0.7, 1, 1.5, and 2) using Weka software tool. The below table is the final normalized data set contains 495 samples with a different set of probes for different threshold values.

Table 1: Representation of the number of probes and log-fold change values for model-1.

Model-1	
LogFc	No. of genes
0.5	2074
0.7	821
1	213
1.5	24
2	3

Table 2: Representation of the number of probes and log-fold change (LogFc) values for model-2.

Model-2	
LogFC	No. of genes
0.5	3513
0.7	2028
1	836
1.5	186
2	52

Table 3: Name and definition of performance evaluation metrics.

Name of Metric	Definition
True Positive (TP)	Correctly predicted sand boil images
True Negative (TN)	Correctly predicted sand boil images
False Positive (FP)	Incorrectly predicted sand boil images
False Negative (FN)	Incorrectly predicted sand boil images
Recall/Sensitivity (Sens.) /True Positive Rate (TPR)	$\frac{TP}{TP+FN}$
Specificity (Spec.) /True Negative Rate (TNR)	$\frac{TN}{TN+FP}$
Fall Out Rate (FOR) /False Positive Rate (FPR)	$\frac{FP}{FP+TN}$
Miss Rate (MR) /False Negative Rate (FNR)	$\frac{FN}{FN+TP}$
Accuracy (ACC)	$\frac{TP+TN}{FP+TP+TN+FN}$
Balanced Accuracy (BACC)	$\frac{1}{2}\left(\frac{TP}{TP+FN}+\frac{TN}{TN+FP}\right)$
Precision (Prec.)	$\frac{TP}{TP+FP}$
F1 score (Harmonic mean of precision and recall)	$\frac{2TP}{2TP+FP+FN}$
Mathews Correlation Coefficient (MCC)	$\frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FN)\times(TP+FP)\times(TN+FP)\times(TN+FN)}}$

4.1 Stacking

The idea of stacking based machine learning technique [33] has recently been successfully applied to solve bioinformatics and computer vision problems [34-39]. Stacking is a model, which obtains information from multiple different models and aggregates them to obtain a new model.

The generalized error rate will be minimized and yields to more accurate results when the information is gained from more than one predictive model.

There are two stages of learners in stacking. The first stage of classifiers is known as base classifiers, and the second stage of classifiers are considered as meta classifiers. In stacking, more than one classifier is used in the first stage as base classifiers. The generalized error rate is reduced by combining the prediction probabilities from the base classifiers using a meta-classifier. To supply the meta classifier with complementary clues, the classifiers in the first stage (base classifiers) must be different from one another based on their operating principles.

To find the meta classifiers and base classifiers to use in the second and first stages of the stacking framework. we examined nine different machine learning algorithms:

(h) Support vector machine (SVM).

(i) Logistic Regression (LogReg).

(j) Random Decision Forest (RDF).

(k) Extra Tree Classifier (ETC).

(l) Gradient Boosting Classifier (GBC).

(m) K Nearest Neighbor (KNN).

(n) eXtreme Gradient Boosting (XGBoost).

We examined four different stacking models. The mentioned stacking models are built and optimized using Scikit-learn [40]. We used 2 sets of datasets (dataset with All-Samples and dataset with All-Samples except for Gleason grade: 7) in stacking.

The below-mentioned stacking models are performed using a dataset with All-Samples (Gleason grade: 6, 7, 8, 9, 10):

i. LogReg, KNN, SVM as the base classifiers, SVM as the meta classifier.

ii. LogReg, KNN, SVM as the base classifiers, XGBC as the meta classifier.

iii. LogReg, KNN, SVM, XGBC as the base classifiers, XGBC as the meta classifier.

iv. RDF, LogReg, KNN as the base classifiers, GBC as the meta classifier.

v. RDF, LogReg, GBC as the base classifiers, KNN as the meta classifier.

The below-mentioned stacking models are performed using a dataset with All-Samples except:7 (Gleason grade: 6, 8, 9, 10):

i. LogReg, KNN, SVM as the base classifier, SVM as the meta classifier.

ii. LogReg, KNN, SVM as the base classifier, XGBC as the meta classifier.

iii. LogReg, KNN, SVM, XGBC as the base classifier, XGBC as the meta classifier.

iv. RDF, LogReg, KNN as the base classifier, GBC as the meta classifier.

v. RDF, LogReg, GBC as base classifier, KNN as meta classifier.

Chapter 5 – Results and Discussions

In this section, We used Principal component analysis to check the misclassified instances in our normalized data-set. The machine learning approach is also implemented for classifying the misclassified samples. We assume that the misclassification rate is high in samples with GG: 7. Initially, the accuracy of our normalized dataset was around 75%, and Using weka, we classified most of the samples and improved accuracy to around 85%. Moreover, We implemented stacking techniques using different combinations of classifiers (a few of them as base classifiers and few as meta classifier) and improved the accuracy and reduced the error rate.

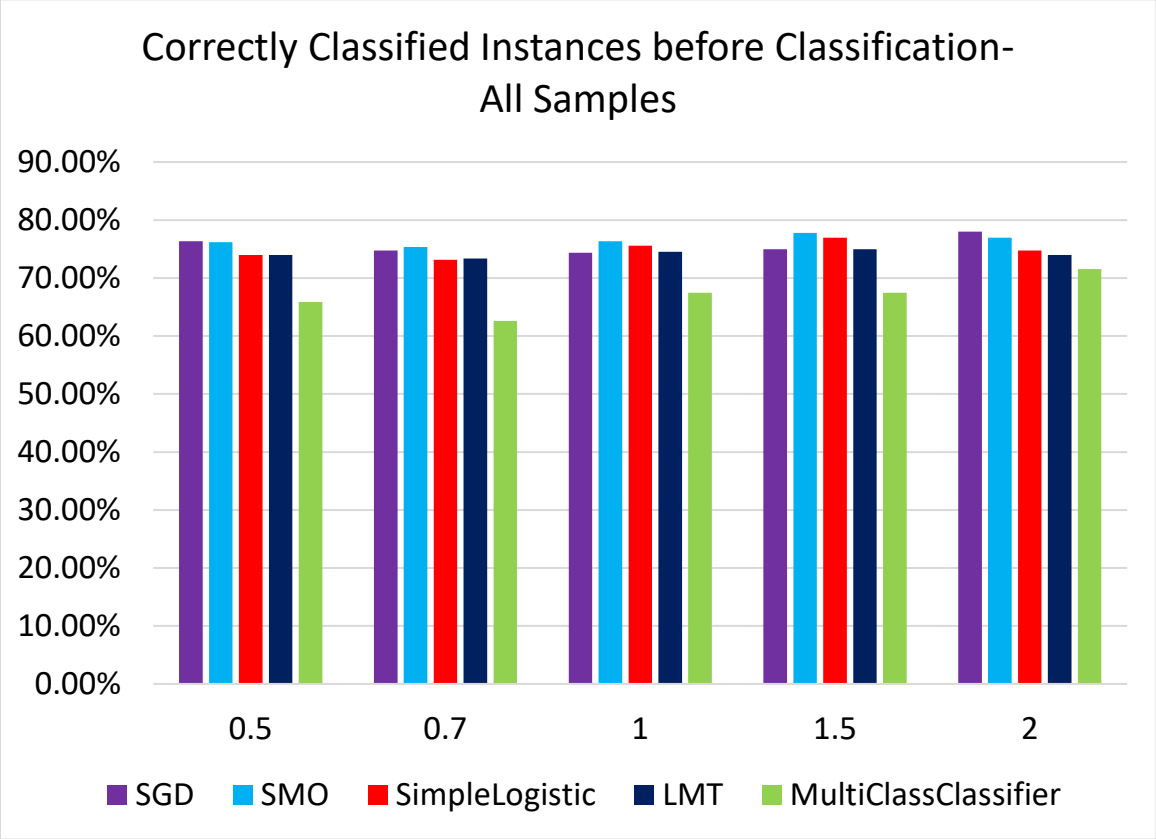


Figure 5.1: Figure showing the accuracy percentage of all the samples with Gleason Grade: 6, 7, 8, 9, 10 before classification.

Indolent: Red

Aggressive: Blue

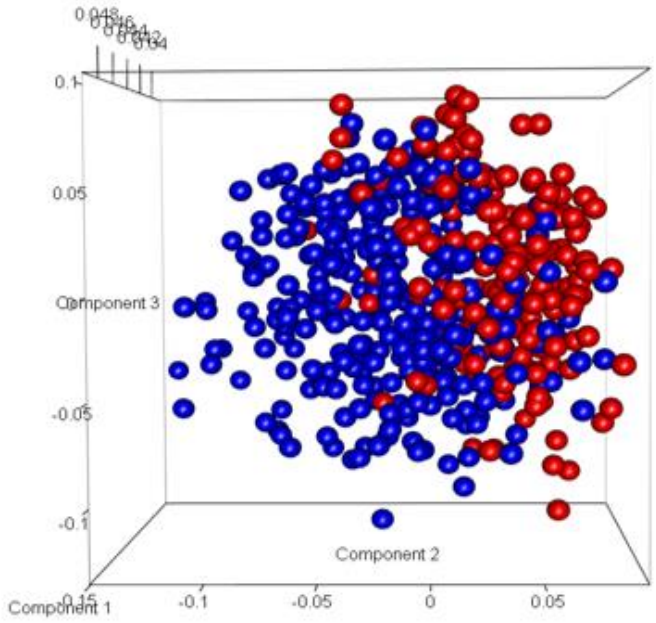


Figure 5.2: Principal component analysis on normalized data with all the samples (Gleason grade: **6, 7, 8, 9, 10**).

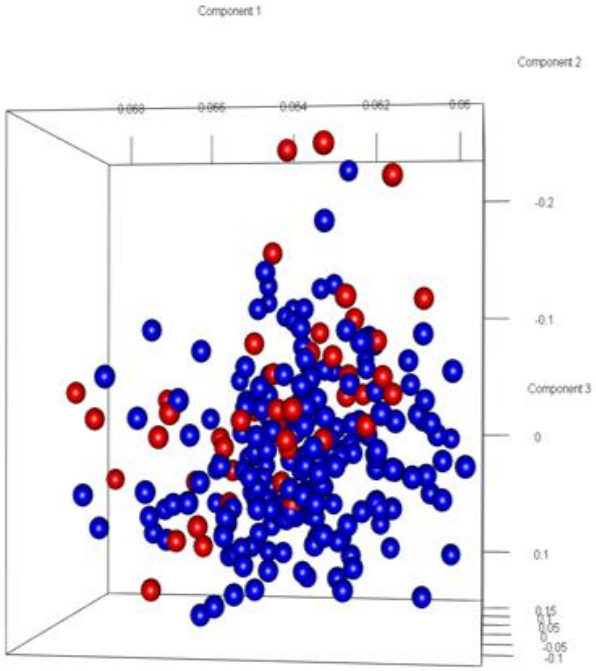


Figure 5.3: Principal component analysis on normalized data with all the samples except GG: 7 (Gleason Score: **6, 8, 9, 10**).

Figure 5.6 represents the percentage of correctly classified instances for all the samples using 5 different classifiers, and Figure 5.2 and Figure 5.3 contain the principal component analysis of 2 data sets before classification. One is the principal component of all samples (495) with Gleason Grade: 6, 7, 8, 9, and 10 containing 2074 probes and the other is the principal component analysis of the samples (249) with Gleason Grade: 6, 8, 9, and 10 containing 3513 probes. As we can see, the samples (Indolent and Aggressive) are mixing a little bit. We assumed this case as Model 1 and applied machine learning approach.

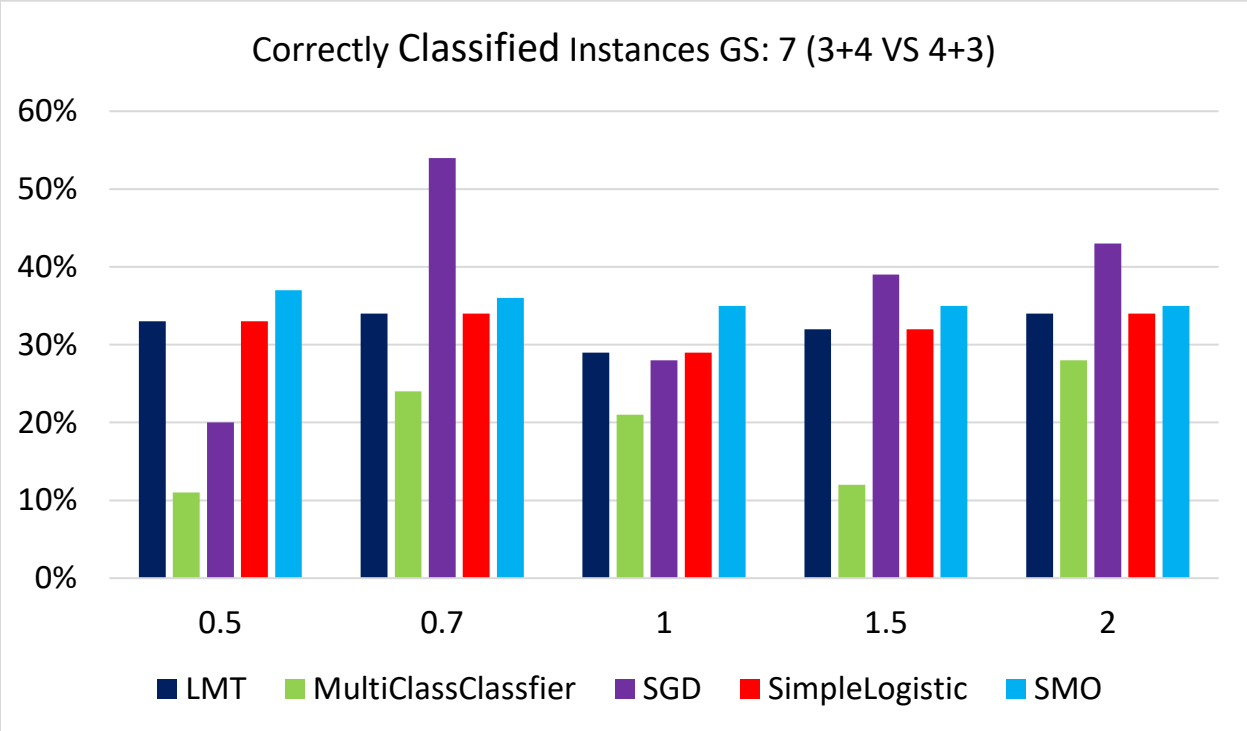


Figure 5.4: Figure showing the accuracy of samples with GG: 7 (3+4 and 4+3).

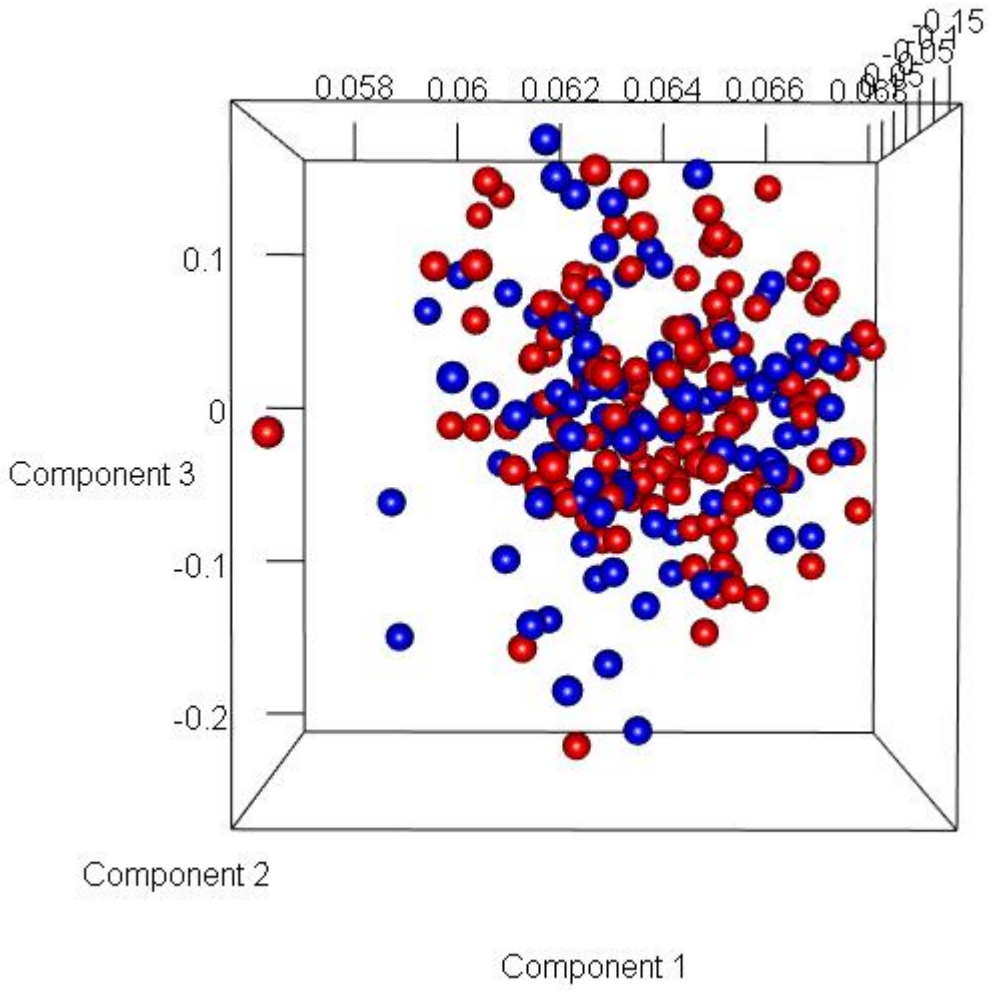
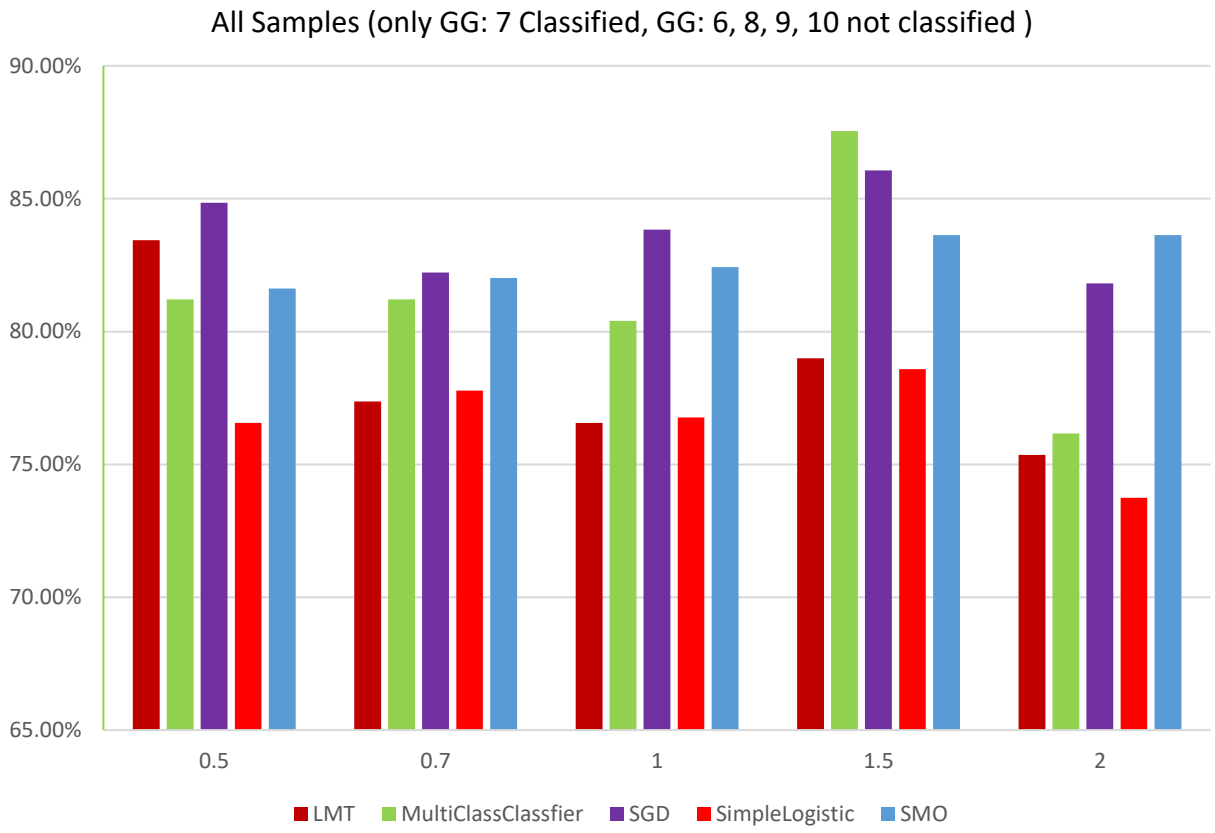


Figure 5.5: Principal component analysis of samples with GG: 7 in 3-dimension.

Figure 5.6 represents the percentage of correctly classified instances in the normalized data (Indolent Vs. Aggressive) with the samples containing 7, which consists of 3+4 (primary grade as 3 and secondary grade as 4) or 4+3 (primary grade as 4 and secondary grade as 3) and Figure 5.5 represents the principal component analysis of samples with Gleason Grade 7 (3+4 and 4+3) in 3-dimension.



GG: Gleason grade

Figure 5.6: Figure represents the accuracy of all the samples after classifying only samples with GG: 7 for 5 different classifiers with different log-fold change values.

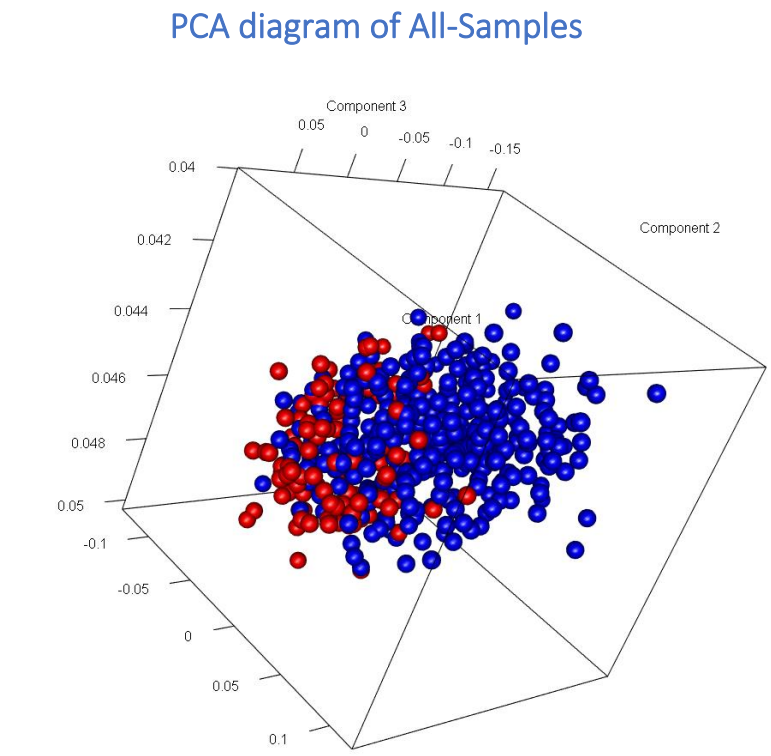


Figure 5.7: Principal component analysis of LMT classifier of all samples in 3-dimension with Gleason Grade: 6, 7, 8, 9, and 10.

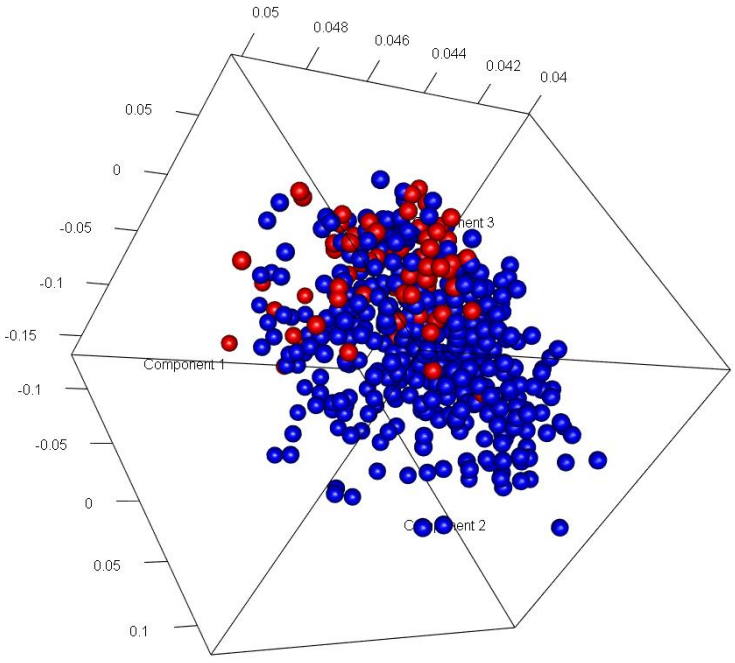


Figure 5.8: Principal component analysis of MultiClassClassifier classifier of all samples in 3-dimension with Gleason Grade: 6, 7, 8, 9, and 10.

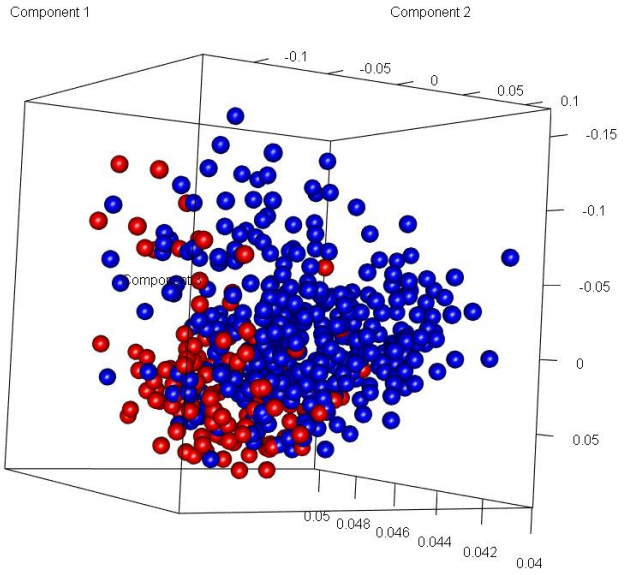


Figure 5.9: Principal component analysis of SGD classifier of all samples in 3-dimension with Gleason Grade: 6, 7, 8, 9, and 10.

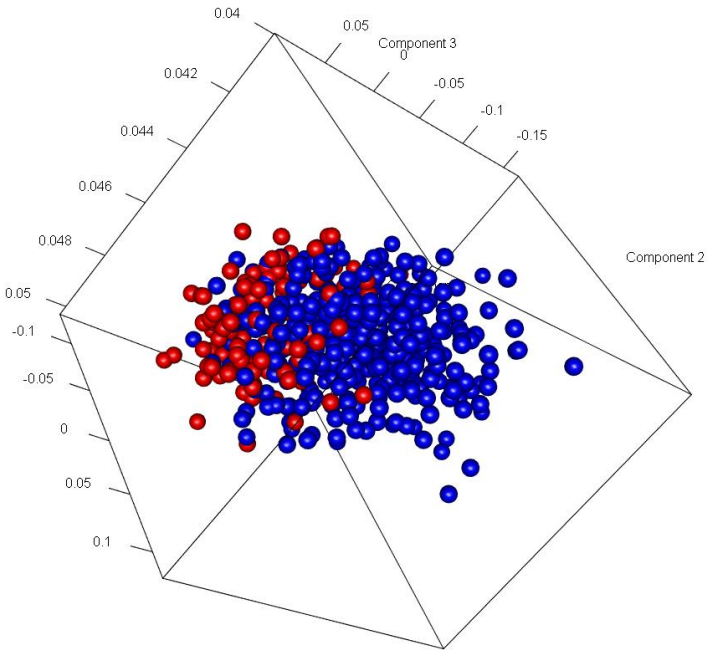


Figure 5.10: Principal component analysis of SimpleLogistic classifier of all samples in 3-dimension with Gleason Grade: 6, 7, 8, 9, and 10.

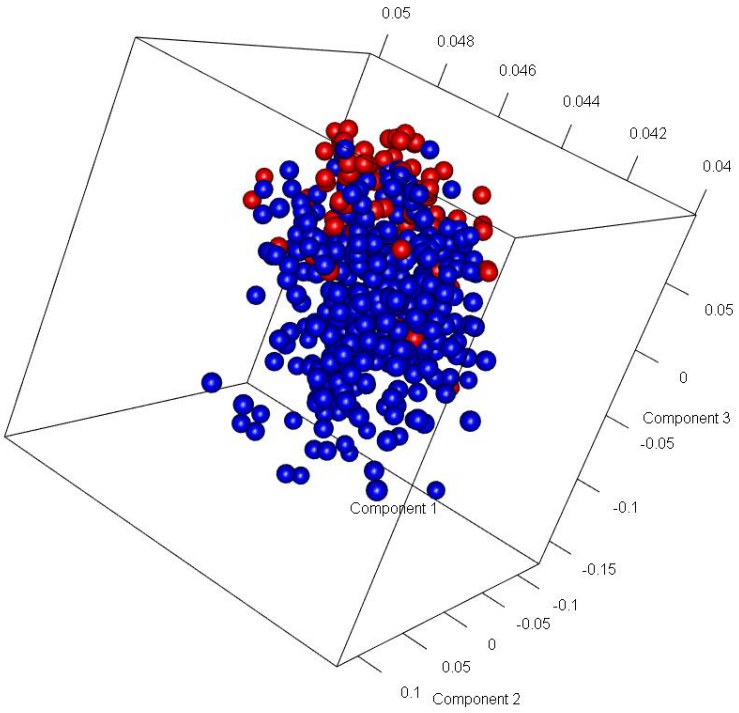


Figure 5.11: Principal component analysis of SMO classifier of all samples in 3-dimension with Gleason Grade: 6, 7, 8, 9, and 10.

The above figures from Figure 5.7 to Figure 5.11 are the principal component analysis of all the samples (495) obtained by applying machine learning using 5 different classifiers after classifying the misclassified samples in the normalized data with the threshold (1.5). As the above Figure 5.6 shows, the percentage of correctly classified instances is high for threshold (1.5). There are few samples still mix with others. The reason behind not finding all misclassified instances in GG: 7 is because our normalized data-set with samples (GG: 6, 8, 9, 10) contains few misclassified instances.

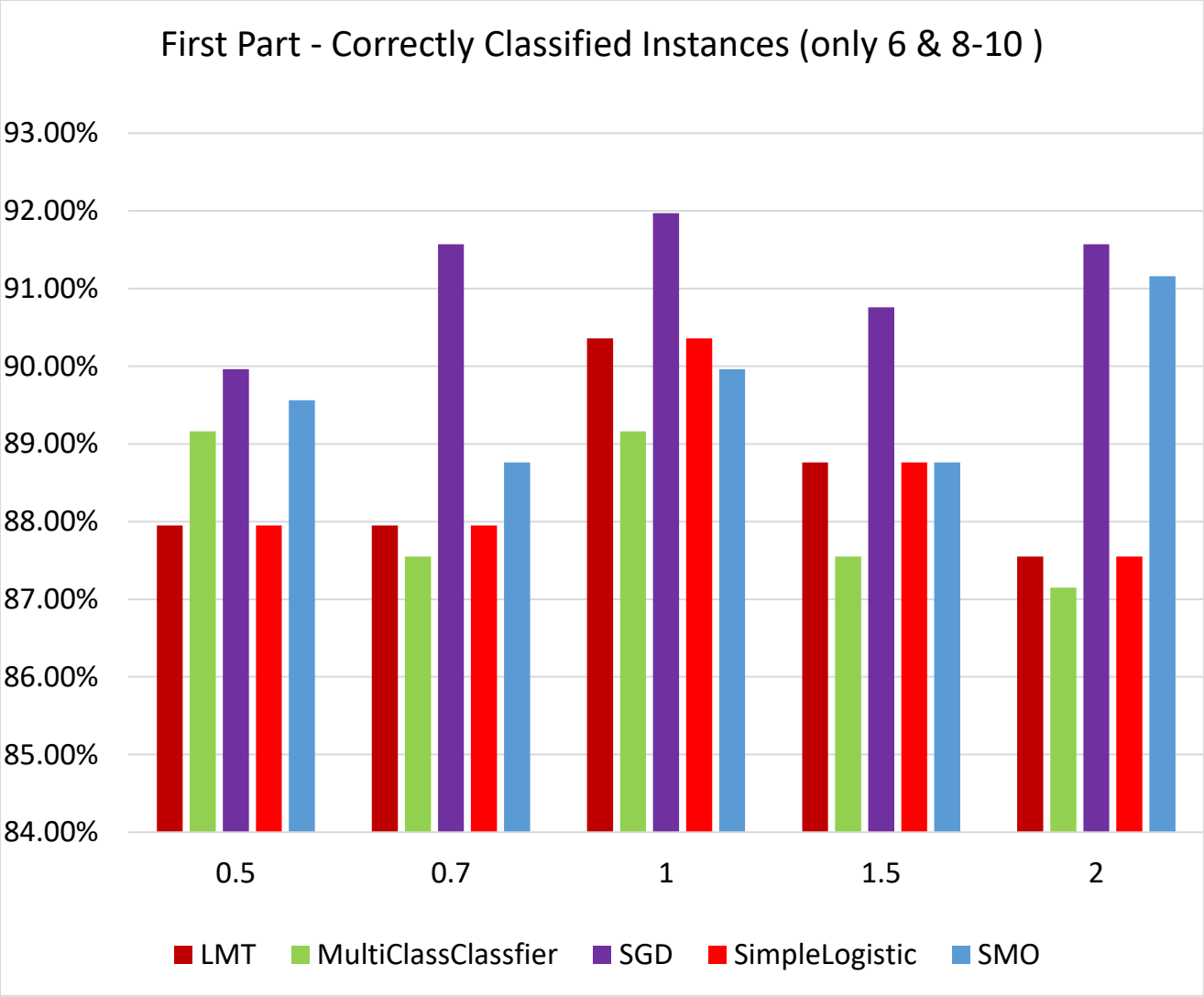


Figure 5.12: The figure represents the accuracy of all samples (with GG: 6, 8, 9, 10) for 5 different classifiers with different log-fold change values.

PCA of ALL-Samples except for GG: 7

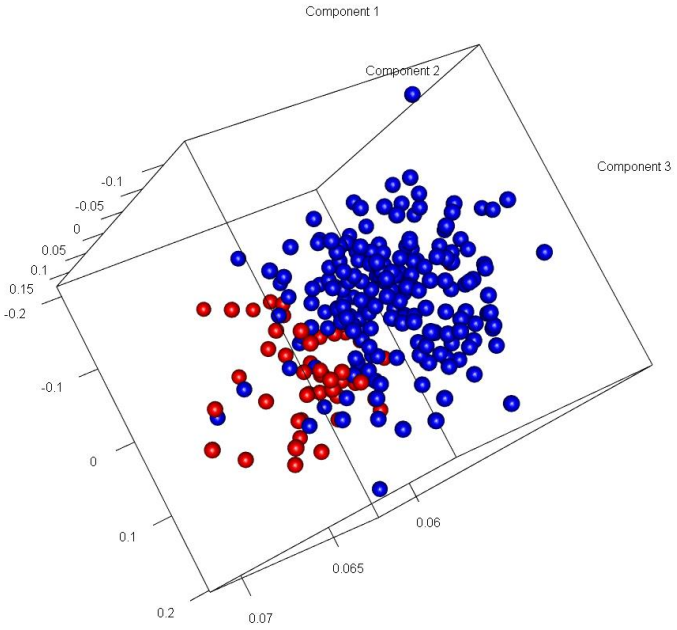


Figure 5.13: Principal component analysis of LMT classifier of samples with Gleason Grade: 6, 8, 9, and 10 in 3-dimension.

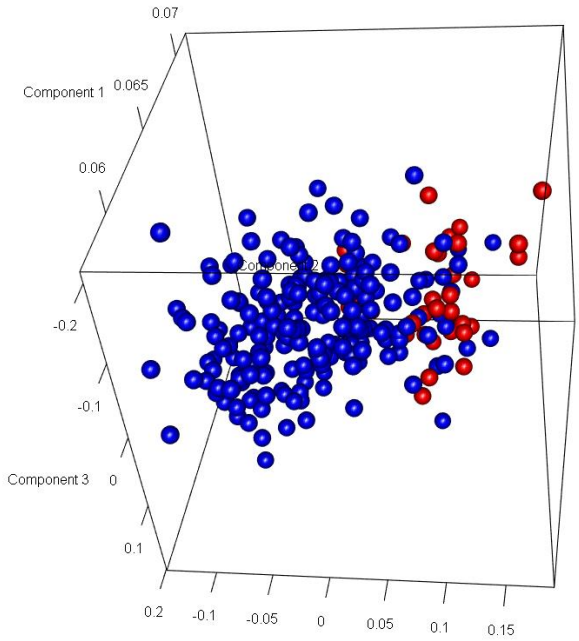


Figure 5.14: Principal component analysis of MultiClassClassifier classifier of samples with Gleason Grade: 6, 8, 9, and 10 in 3-dimension.

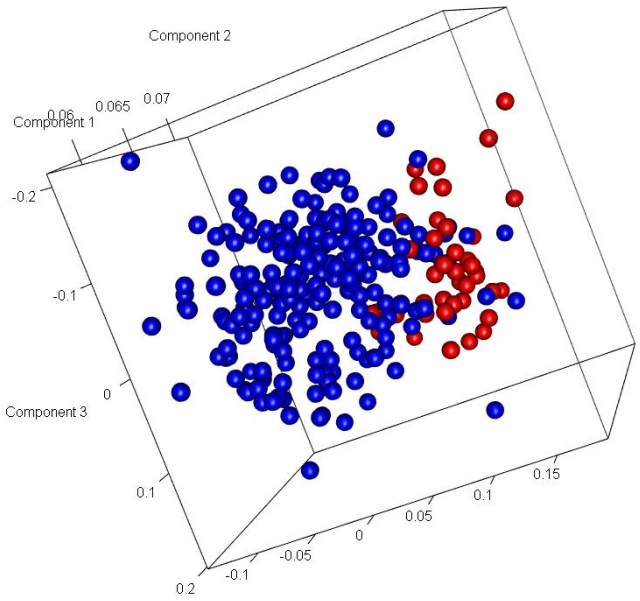


Figure 5.15: Principal component analysis of SGD classifier of samples with Gleason Grade: 6, 8, 9, and 10 in 3-dimension.

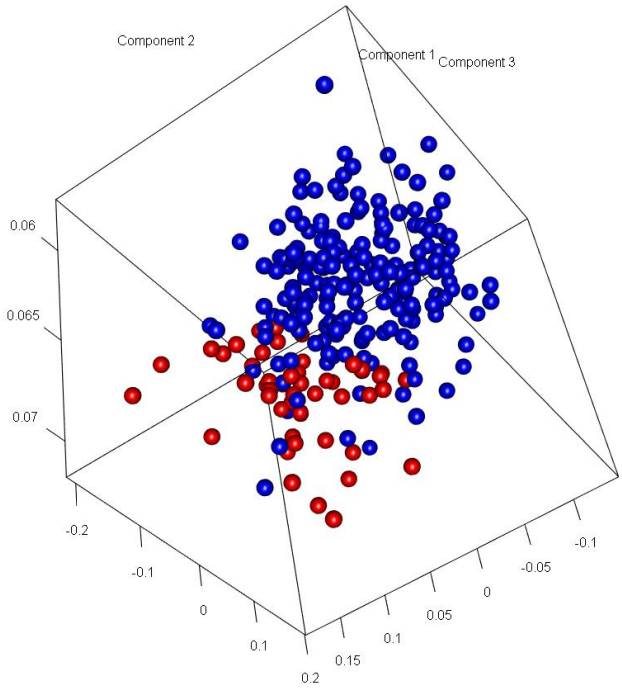


Figure 5.16: Principal component analysis of SimpleLogistic classifier of samples with Gleason Grade: 6, 8, 9, and 10 in 3-dimension.

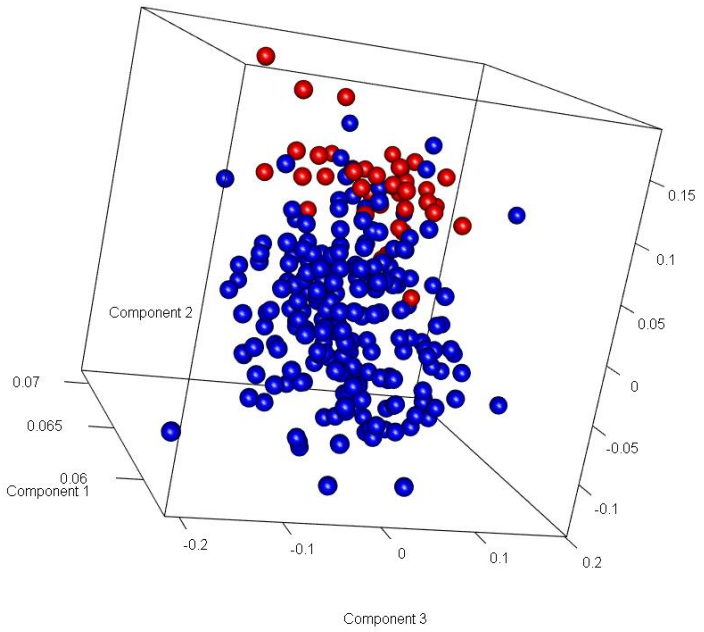


Figure 5.17: Principal component analysis of SMO classifier of samples with Gleason Grade: 6, 8, 9, and 10 in 3-dimension.

The above figures from Figure 5.13 to Figure 5.17 are the principal component analysis of the samples (249) with Gleason Grade: (6, 8, 9, 10) obtained by applying machine learning using 5 different classifiers after classifying the misclassified samples in the normalized data with the threshold value: 1 and by eliminating the samples with Gleason Grade: 7. From the above Figure 5.12, the percentage of correctly classified instances is high for threshold value: 1 in normalized data set.

5.1 Stacking

We tried different combinations of base classifiers and meta classifiers in the stacking technique. These classifiers were chosen based on different principles and their accuracies independently. The below tables shows the comparison of these stacking models. The performance of the stacking technique depends on the principles that each of base classifiers

helps the meta-learner to perform better. In our case, the model performs the best with better accuracy with data set containing all the samples with Gleason grade: 6, 7, 8, 9, and 10 and the model performs the best with the best accuracy with data set containing all the samples except Gleason grade: 7 (samples with Gleason grade: 6, 8, 9, and 10). Model and have almost a similar accuracy with all the samples with Gleason grade: 6, 7, 8, 9, and 10 and Model and have almost a similar accuracy with all the samples except Gleason grade: 7 (samples with Gleason Grade: 6, 8, 9, 10).

Table 4: Performance of various Classifiers with data set containing all the samples with Gleason grade: 6, 7, 8, 9, and 10.

	Model type and Description	Sensitivity	Specificity	Accuracy	Precision	F1 Score	MCC	Balanced Accuracy
I.	Suppert Vector Machine (SVM)	0.91351	0.68800	**0.85657**	0.89655	0.90495	0.61332	0.80076
II.	Logistic Regression (LogReg)	0.84865	0.67200	0.80404	0.88451	0.86621	0.50225	0.76032
III.	Random Decision Forest (RDF)	0.92432	0.51200	0.82020	0.84864	0.88486	0.48733	0.71286
IV.	Extra Tree Classifier (ETC)	0.92703	0.48800	0.81616	0.84275	0.88288	0.84275	0.71946
V.	Gradient Boosting Classifier (GBC)	0.91081	0.54400	0.81818	0.85533	0.88220	0.49032	0.71941
VI.	K nearest neighbor (KNN)	0.85676	0.59200	0.78990	0.86141	0.85908	0.44642	0.72438
VII.	eXtreme Gradient Boosting (XGBC)	0.90210	0.66892	0.85051	0.88761	0.88914	0.60723	0.79830

Table 5: Performance of various Stacking methods with data set containing all the samples with Gleason grade: 6, 7, 8, 9, and 10.

Model type and Description		Sensitivity	Specificity	Accuracy	Precision	F1 Score	MCC	Balanced Accuracy
I.	LogReg, KNN, SVM as Base, SVM as Meta-classifier	0.99198	0.85124	**0.95758**	0.95373	0.97248	0.88337	0.92161
II.	LogReg, SVM, KNN, XGBC as Base, XGBC as Meta-classifier	0.95989	0.83471	0.92929	0.94723	0.95352	0.80618	0.87295
III.	LogReg, KNN, SVM as Base, XGBC as Meta-classifier	0.972	0.8368	0.93131	0.949	0.961	0.8369	0.90690
IV.	RDF, LogReg, KNN as Base, GBC as Meta-classifier	0.98396	0.67769	0.90909	0.90418	0.94238	0.74373	0.83082
V.	RDF, LogReg, GBC as Base, KNN as Meta-classifier	0.93583	0.80992	0.90505	0.93834	0.93708	0.74368	0.87287

Table 6: Performance of various Stacking methods with data set containing all the samples with Gleason grade: 6, 8, 9, and 10.

Model type and Description		Sensitivity	Specificity	Accuracy	Precision	F1 Score	MCC	Balanced Accuracy
I.	LogReg, KNN, SVM as Base, SVM as Meta-classifier	0.98182	0.89655	**0.97189**	0.98630	0.98405	0.86558	0.93918
II.	LogReg, SVM, KNN, XGBC as Base, XGBC as Meta-classifier	0.95127	0.7911	0.93812	0.96976	0.95991	0.71929	0.90869
III.	LogReg, KNN, SVM as Base, XGBC as Meta-classifier	0.95909	0.79310	0.94779	0.97235	0.96568	0.72100	0.91513
IV.	RDF, LogReg, KNN as Base, GBC as Meta-classifier	0.97727	0.62069	0.93574	0.95133	0.96413	0.66247	0.79898
V.	RDF, LogReg, GBC as Base, KNN as Meta-classifier	0.97727	0.58621	0.93173	0.94714	0.96197	0.63689	0.78174

Chapter 6 – Conclusions

In this thesis, we compared different classifiers and determined the best one to use for classifying prostate cancer patients. We also implemented a stacking-based machine learning technique to increase the prediction accuracy of the machine learning model using a different combination of classifiers. Yes, the results were improved by using a stacking based machine learning technique. The accuracy was increased from ~85.657% to ~95.758%.

We also used the Genetic Algorithm method for feature elimination on both the data sets (samples with GG: 6, 7, 8, 9, 10, and samples with GG: 6, 8, 9, 10). The number of genes obtained for samples with GG: 6, 7, 8, 9, and 10 is 1020. The number of genes obtained for samples with GG: 6, 8, 9, and 10 is 1681. The best fitness values for all log-fold change values (0.5, 0.7, 1, 1.5, 2) for both data sets were obtained using the Genetic Algorithm. The fitness value oftained for samples with GG: 6, 7, 8, 9, and 10 is 1.6140201. and the fitness value obtained for samples with GG: 6, 8, 9, and 10 is 1.74722.The highest accuracy obtained for our data set before applying machine learning is about 75% for all classifiers. We also used the Weka software [41] tool for classifying prostate cancer patients using different classifiers individually, and the highest accuracy obtained by Weka is about 85% for different classifiers individually.

Furthermore, we tried with few other classifiers separately and found that SVM performs the best 10-fold-cross-validation, achieving high accuracy of 86.465%, and XGBoost was also performing with high accuracy of 85.051%. Moreover, stacking on all machine learning methods revealed an even better performance of 95.758% accuracy for data set containing GG: 6, 7, 8, 9, and 10, and 97.19% accuracy for the data set containing GG: 6, 8, 9, and 10. Hence, we found

that the samples with GG:7 are more misclassified compared to samples with GG: 6, 8, 9, and 10. The stacking machine learning technique might prove to be useful in classifying prostate cancer patients. In order to improve the accuracy or classify all the patients correctly, mutation-based or methylation-based analysis can be implemented to yield better results in the classification of prostate cancer patients.

References

[1] S. Rodney, T. T. Shah, H. R. Patel, and M. Arya, "Key papers in prostate cancer," *Expert Rev Anticancer Ther,* vol. 14, pp. 1379-84, Nov 2014.

[2] M. J. Watson, A. K. George, M. Maruf, T. P. Frye, A. Muthigi, M. Kongnyuy*, et al.*, "Risk stratification of prostate cancer: integrating multiparametric MRI, nomograms and biomarkers," *Future oncology (London, England),* vol. 12, pp. 2417-2430, 2016.

[3] J. I. Epstein, W. C. Allsbrook, Jr., M. B. Amin, and L. L. Egevad, "The 2005 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma," *Am J Surg Pathol,* vol. 29, pp. 1228-42, Sep 2005.

[4] J. I. Epstein, L. Egevad, M. B. Amin, B. Delahunt, J. R. Srigley, and P. A. Humphrey, "The 2014 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma: Definition of Grading Patterns and Proposal for a New Grading System," *Am J Surg Pathol,* vol. 40, pp. 244-52, Feb 2016.

[5] A. Lavi and M. Cohen, "[PROSTATE CANCER EARLY DETECTION USING PSA - CURRENT TRENDS AND RECENT UPDATES]," *Harefuah,* vol. 156, pp. 185-188, Mar 2017.

[6] V. A. Moyer, "Screening for Prostate Cancer: U.S. Preventive Services Task Force

[7] K. Lin, J. M. Croswell, H. Koenig, C. Lam, and A. Maltz, "U.S. Preventive Services Task Force Evidence Syntheses, formerly Systematic Evidence Reviews," in *Prostate-Specific Antigen-Based Screening for Prostate Cancer: An Evidence Update for the U.S. Preventive Services Task Force*, ed Rockville (MD): Agency for Healthcare Research and Quality (US), 2011.

[8] S. S. Sayantan mitra, Sudipta Acharya, "Fusion of stability and multi-objective optimization for solving cancer tissue classification problem," *Expert Systems with Applications,* vol. 113, pp. 377-396, 2018.

[9] J. Cuzick, G. P. Swanson, G. Fisher, A. R. Brothman, D. M. Berney, J. E. Reid*, et al.*, "Prognostic value of an RNA expression signature derived from cell cycle proliferation genes in patients with prostate cancer: a retrospective study," *The Lancet. Oncology,* vol. 12, pp. 245-255, 2011.

[10] F. Mo, D. Lin, M. Takhar, V. R. Ramnarine, X. Dong, R. H. Bell*, et al.*, "Stromal Gene Expression is Predictive for Metastatic Primary Prostate Cancer," *Eur Urol,* vol. 73, pp. 524-532, Apr 2018.

[11] K. L. Penney, J. A. Sinnott, K. Fall, Y. Pawitan, Y. Hoshida, P. Kraft*, et al.*, "mRNA expression signature of Gleason grade predicts lethal prostate cancer," *J Clin Oncol,* vol. 29, pp. 2391-6, Jun 10 2011.

[12] P. Danaee, R. Ghaeini, and D. A. Hendrix, "A DEEP LEARNING APPROACH FOR CANCER DETECTION AND RELEVANT GENE IDENTIFICATION," *Pac Symp Biocomput,* vol. 22, pp. 219-229, 2017.

[13] G. Golcuk, M. A. Tuncel, and A. Canakoglu, "Exploiting Ladder Networks for Gene Expression Classification," in *Bioinformatics and Biomedical Engineering*, Cham, 2018, pp. 270-278.

[14] L. Yang, S. Wang, M. Zhou, X. Chen, W. Jiang, Y. Zuo*, et al.*, "Molecular classification of prostate adenocarcinoma by the integrated somatic mutation profiles and molecular network," *Scientific reports,* vol. 7, pp. 738-738, 2017.

[15] T. Takeuchi, M. Hattori-Kato, Y. Okuno, S. Iwai, and K. Mikami, "Prediction of prostate cancer by deep learning with multilayer artificial neural network," *Canadian Urological Association journal = Journal de l'Association des urologues du Canada,* vol. 13, pp. E145-E150, 2019.

[16] M. Casey, B. Chen, J. Zhou, and N. Zhou, "A Machine Learning Approach to Prostate Cancer Risk Classification Through Use of RNA Sequencing Data," in *Big Data – BigData 2019*, Cham, 2019, pp. 65-79.

[17] J. S. Dudani, M. Ibrahim, J. Kirkpatrick, A. D. Warren, and S. N. Bhatia, "Classification of prostate cancer using a protease activity nanosensor library," *Proc Natl Acad Sci U S A,* vol. 115, pp. 8954-8959, Sep 4 2018.

[18] L. Spahić and S. Ćordić, "Prostate Tissue Classification Based on Prostate-Specific Antigen Levels and Mitochondrial DNA Copy Number Using Artificial Neural Network," in *CMBEBIH 2019*, Cham, 2020, pp. 649-654.

[19] W. Zhou, M. Zhu, M. Gui, L. Huang, Z. Long, L. Wang*, et al.*, "Peripheral blood mitochondrial DNA copy number is associated with prostate cancer risk and tumor burden," *PLoS One,* vol. 9, p. e109470, 2014.

[20] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Transactions on Neural Networks,* vol. 10, pp. 988-999, 1999.

[21] A. Szilagyi and J. Skolnick, "Efficient prediction of nucleic acid binding function from low-resolution protein structures," *J Mol Biol,* vol. 358, pp. 922-33, May 5 2006.

[22] H. Tin Kam, "Random decision forests," in *Proceedings of 3rd International Conference on Document Analysis and Recognition*, 1995, pp. 278-282 vol.1.

[23] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning,* vol. 63, pp. 3-42, 2006/04/01 2006.

[24] J. H. Friedman, "Greedy Function Approximation: A Gradient Boosting Machine," *The Annals of Statistics,* vol. 29, pp. 1189-1232, 2001.

[25] N. S. Altman, "An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression," *The American Statistician,* vol. 46, pp. 175-185, 1992/08/01 1992.

[26] T. Chen and C. Guestrin, *XGBoost: A Scalable Tree Boosting System*, 2016.

[27] N. Landwehr, M. Hall, and E. Frank, "Logistic Model Trees," *Machine Learning,* vol. 59, pp. 161-205, 2005/05/01 2005.

[28] N. C. Institute, "GDC Data Portal."

[29] N. Chen and Q. Zhou, "The evolving Gleason grading system," *Chinese journal of cancer research = Chung-kuo yen cheng yen chiu,* vol. 28, pp. 58-64, 2016.

[30] A. T. Belinda Phipson, Matt Ritchie, Maria Doyle, Harriet Dashnow, Charity Law, "RNA-seq analysis in R," 2016.

[31] J. Brownlee, "How to Run Your First Classifier in Weka," *Machine Learning Mastery,* 2014.

[32] M. D. Robinson and A. Oshlack, "A scaling normalization method for differential expression analysis of RNA-seq data," *Genome Biology,* vol. 11, p. R25, 2010/03/02 2010.

[33] D. H. Wolpert, "Stacked generalization," *Neural Networks,* vol. 5, pp. 241-259, 1992/01/01/ 1992.

[34] S. Iqbal and M. Hoque, "PBRpredict-Suite: A Suite of Models to Predict Peptide Recognition Domain Residues from Protein Sequence," *Bioinformatics (Oxford, England),* vol. 34, 05/03 2018.

[35] A. Mishra, P. Pokhrel, and M. Hoque, "StackDPPred: A Stacking based Prediction of DNA-binding Protein from Sequence," *Bioinformatics,* vol. 35, 07/19 2018.

[36] Q. Hu, C. Merchante, A. N. Stepanova, J. M. Alonso, and S. Heber, "A Stacking-Based Approach to Identify Translated Upstream Open Reading Frames in Arabidopsis Thaliana," in *Bioinformatics Research and Applications*, Cham, 2015, pp. 138-149.

[37] S. Gattani, A. Mishra, and M. T. Hoque, "StackCBPred: A stacking based prediction of protein-carbohydrate binding sites from sequence," *Carbohydr Res,* vol. 486, p. 107857, Dec 1 2019.

[38] D. S. M. A. Aditi S. Kuchi, Dr. Minhaz F. Zibran, Dr. Mahdi Abdelguerfi, Dr. Md Tamjidul Hoque, "Detection of Sand Boils from Images using Machine Learning Approaches," *ScholarWorks@UNO*.

[39] M. Flot, A. Mishra, A. S. Kuchi, and M. T. Hoque, "StackSSSPred: A Stacking-Based Prediction of Supersecondary Structure from Sequence," *Methods in molecular biology (Clifton, N.J.),* vol. 1958, pp. 101-122, 2019 2019.

[40] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel*, et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research,* vol. 12, 01/02 2012.

[41] Z. Markov and I. Russell, *An introduction to the WEKA data mining system* vol. 38, 2006.

Vita

The author, Yashwanth Karthik Kumar Mamidi, was born in Telangana, India. He obtained his bachelor's degree in 2018 from Gitam University, Hyderabad, India. He joined the University of New Orleans Computer Science Master's program in Fall-2018. He worked as a graduate research assistant under Dr. Md Tamjidul Hoque, in his Bioinformatics and Machine Learning Lab (BML Lab) within the Department of Computer Science, at the University of New Orleans. Moreover, he did his internship at Louisiana State University Health Science Center under Dr. Chindo Hicks. He worked on the classification of Prostate cancer patient into Indolent and Aggressive using machine learning as part of his Master thesis in computer science.