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

Yashwanth Karthik Kumar Mamidi

University of New Orleans, Louisiana, USA

ymamidi@my.uno.edu

Saiful Ratul

University of New Orleans, Louisiana, USA

sratul@uno.edu

***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 Score (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 used MRMR to extract features and to overcome with the problem class-imbalance. Then, we implemented SMOTE technique to make class-balanced and improved the accuracy.**

***Keywords* – *ML, MRMR, SMOTE, PROSTATE CANCER, GLEASON SCORE.***

1. Introduction

Prostate cancer (PCa) is the most common solid tumor and the second most common cause of cancer death in the United States [15]. To date, treatment decisions for PCa patients are guided by various risk stratification algorithms [19]. 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 score (GS) [5, 6]. The GS ranges from 6 to 10. The majority of PCa present GS 6. These cancers are associated with very low cancer-specific mortality rates, even in the absence of therapy. Intermediated grade PCa presents GS 7. These cancers present a much more variable clinical course. Localized high grade (aggressive) with lethal potential PCa presents GS: 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 [11]. 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 [13]. 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 [12]. 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 PCa 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). Figure 1 shows the classification Algorithm of Gleason Score.

A screenshot of a cell phone

Description automatically generated

Fig. 1. Classification Algorithm (Gleason Score).

1. Machine Learning Methods

In this Section, we discuss about the different machine learning algorithms and their respective principles used in our project. They are: SVM, LogReg, RDF, ETC, GBC, KNN, Hoeffding, MultiClassClassifier, LMT, NaiveBayes, SimpleLogistic classifiers.

*A. Support Vector Machine (SVM)*

A Support Vector Machine (SVM) [18] 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). 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.

*B. Logistic Regression (LogReg)*

Logistic Regression [16] 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).

*C. Random Decision Forest (RDF)*

Random decision Forest [17] 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 dataset. 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.

*D. Extra Tree Classifier (ETC)*

The Extra Tree [8] 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 dropped 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.

*E. Gradient Boosting Classifier (GBC)*

Gradient boosting classifier [7] 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.

*F. K Nearest Neighbors (KNN)*

K nearest neighbor [1] 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.

A screenshot of a cell phone

Description automatically generated

Fig. 2. Shows the Calculation of distance and finding neighbors and voting for the KNN method.

*G. Hoeffding Classifier*

A Hoeffding classifier is an incremental, anytime decision tree algorithm that is capable of learning from massive data streams, if distribution examples does not change over time. It exploits a fact that a very small sample is enough to choose splitting attribute.

*H. 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.

*I. Logistic Model Trees (LMT)*

The logistic model tree (LMT) [10] 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.

*J. NaiveBayes Classifier*

Naïve Bayes classifier uses estimator classes. Based on analysis of training data, numeric estimator precision values are chosen. This classifier will use a default precision of 0.1 for numeric attributes when buildClassifier is called with zero training instances.

*K. SimpleLogistic Classifier*

SimpleLogistic classifier is used for building linear logistic regression models. LogitBoost with simple regression functions as base learners and is used for fitting the logistic models. The optimal number of LogitBoost iterations to perform is cross validated, which leads to automatic attribute selection.

1. Experimental Materials and Methods

*A. 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 [9], 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 (GS=6), 246 samples with intermediate (GS=7), 204 of aggressive with lethal potential and 52 control samples. Gene expression data used 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 GS because GS =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) score. The samples with GS: 3 + 4 grades were assigned to a group of patients with Gleason Score 6 (Indolent PCa). The samples with GS: 4 + 3 grades were assigned to a group of patients with Gleason Score 8 to 10[3].

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 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.

A close up of a logo

Description automatically generated

## Fig. 3. Project Design

## Data Analysis for Gene Selection

Using the limma and edgeR packages in R 3.8.0[2], 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 multiply to unity across all libraries. Implemented in R before performing statistical tests. The processed normalized data contained 34,956 probes. 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 4 represents the boxplots of logCPM (log counts per million) before normalization.

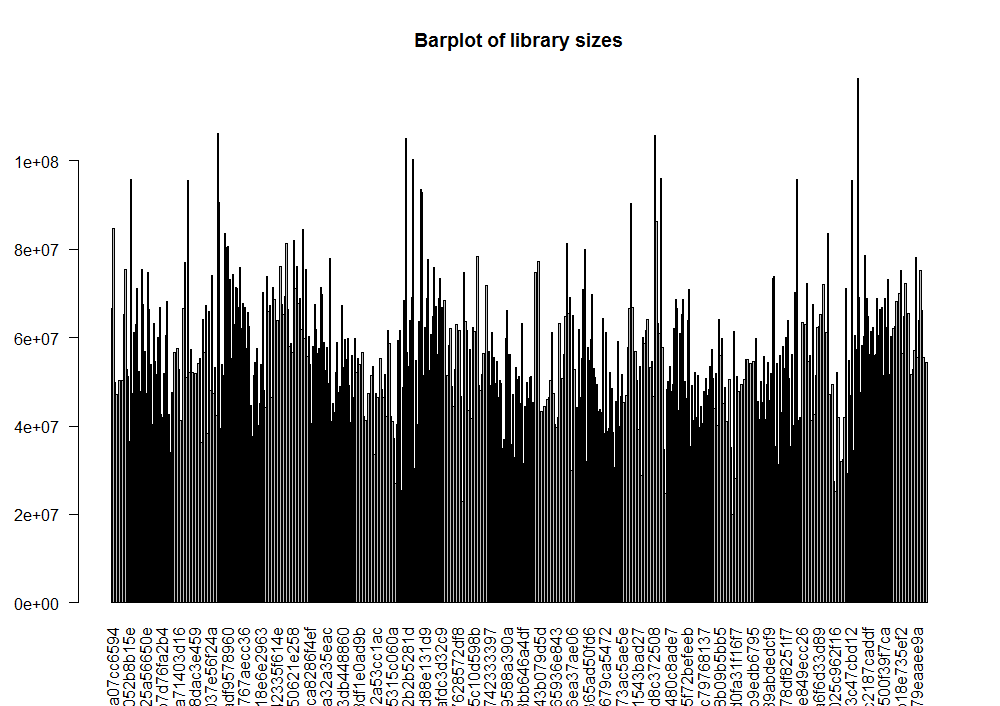


Fig. 4.Library sizes of all samples expressed using a barplot constitutes the data quality and unnormalized library sizes.

1. Maximum Relevance and Minimum Redundancy (MRMR)

MRMR is a feature selection approach in Machine Learning and it tends to select features with a high correlation with the class (output) and low correlation between themselves. Features are selected one by one by applying a greedy search to maximize the objective function. MRMR provides output with ranks for all features. 2074 features are extracted using MRMR feature selection. Objective function is a function of relevance and redundancy. Two commonly used types of the objective function are MID (Mutual Information Difference criterion) and MIQ (Mutual Information Quotient criterion) representing the difference or the quotient of relevance and redundancy, respectively.

1. Synthetic Minority Over-sampling Technique (SMOTE)

SMOTE is a technique used for imbalanced class in a dataset. It generates synthetic samples from the minority class, and It obtains a synthetically balanced class or almost class-balanced training set, which is then used to train different classifiers. Figure 5 clearly shows the dataset is class imbalanced. Using SMOTE technique, the number of samples/class are increased to make the class balanced (Figure 6).

A close up of a map

Description automatically generated

Fig. 5. Unbalanced Data Set

A close up of a map

Description automatically generated

Fig. 6. Balancing Data Set using SMOTE

1. Results

We used machine learning approach for classifying the samples in our datasets. From Figure 7, we can say that our dataset contains accuracy of around 75%. We can clearly say that few samples are misclassified in our dataset. Using SMOTE technique, we increased the number of samples and made the class balanced. We also found few misclassified samples in our dataset using machine learning methods. Figure 8 shows the accuracy percentage of all the samples in our dataset using different type of classifiers. Our accuracies jumped to ~88%.

Fig. 7. Before Classification

Fig. 8. After Classification

1. Conclusion

We compared different classifiers for our dataset using machine learning approach. We also implemented MRMR feature extraction technique to find the features, which contribute the disease to the patients. Later on, we also implemented SMOTE technique for class imbalanced problem. The number of features obtained from MRMR feature extraction technique is 2074. Results show that current classification algorithm could misclassify the prostate cancer patients. Using Machine learning algorithms, we can classify the patients into correct groups (Indolent and Aggressive). We improved the accuracy from ~74% to ~88% using Machine Learning classifiers. In order to classify all samples correctly, Single Cell based analysis or Methylation based analysis can be implemented to yield better results.

References

1. Altman, N. S. (1992), 'An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression', The American Statistician, 46 (3), 175-85.
2. Belinda Phipson, Anna Trigos, Matt Ritchie, Maria Doyle, Harriet Dashnow, Charity Law (2016), 'RNA-seq analysis in R'.
3. Chen, Ni and Zhou, Qiao (2016), 'The evolving Gleason grading system', Chinese journal of cancer research = Chung-kuo yen cheng yen chiu, 28 (1), 58-64.
4. Chen, Tianqi and Guestrin, Carlos (2016), XGBoost: A Scalable Tree Boosting System 785-94.
5. Epstein, J. I., et al. (2005), 'The 2005 International Society of Urological Pathology (ISUP) Consensus Conference on Gleason Grading of Prostatic Carcinoma', Am J Surg Pathol, 29 (9), 1228-42.
6. Epstein, J. I., et al. (2016), '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, 40 (2), 244-52.
7. Friedman, Jerome H. (2001), 'Greedy Function Approximation: A Gradient Boosting Machine', The Annals of Statistics, 29 (5), 1189-232.
8. Geurts, Pierre, Ernst, Damien, and Wehenkel, Louis (2006), 'Extremely randomized trees', Machine Learning, 63 (1), 3-42.
9. Institute, National Cancer 'GDC Data Portal'.
10. Landwehr, Niels, Hall, Mark, and Frank, Eibe (2005), 'Logistic Model Trees', Machine Learning, 59 (1), 161-205.
11. Lavi, A. and Cohen, M. (2017), '[PROSTATE CANCER EARLY DETECTION USING PSA - CURRENT TRENDS AND RECENT UPDATES]', Harefuah, 156 (3), 185-88.
12. Lin, K., et al. (2011), 'U.S. Preventive Services Task Force Evidence Syntheses, formerly Systematic Evidence Reviews', Prostate-Specific Antigen-Based Screening for Prostate Cancer: An Evidence Update for the U.S. Preventive Services Task Force (Rockville (MD): Agency for Healthcare Research and Quality (US)).
13. Moyer, Virginia A. (2012), 'Screening for Prostate Cancer: U.S. Preventive Services Task Force
14. Recommendation Statement', Annals of Internal Medicine.
15. Rodney, S., et al. (2014), 'Key papers in prostate cancer', Expert Rev Anticancer Ther, 14 (11), 1379-84.
16. Szilagyi, A. and Skolnick, J. (2006), 'Efficient prediction of nucleic acid binding function from low-resolution protein structures', J Mol Biol, 358 (3), 922-33.
17. Tin Kam, Ho (1995), 'Random decision forests', Proceedings of 3rd International Conference on Document Analysis and Recognition (1), 278-82 vol.1.
18. Vapnik, V. N. (1999), 'An overview of statistical learning theory', IEEE Transactions on Neural Networks, 10 (5), 988-99.
19. Watson, Matthew J., et al. (2016), 'Risk stratification of prostate cancer: integrating multiparametric MRI, nomograms and biomarkers', Future oncology (London, England), 12 (21), 2417-30.