Prediction of the morality for urinary bladder cancer using machine learning techniques

Khalid Bashir
Fellow, IEEE
University of Engineering and
Technology
Bhaghbanpura
Lahore, TX 12345, Pakistan
khalidbashir1212@gmail.com

***Abstract* – Urinary bladder cancer (UBC) is a common disease in genitourinary malignancy. It is the 5th most common cancer in US, with these high rates of incidence, prediction of survivability for this cancer has still not been given enough attention. The advancement of machine learning techniques have proven to be an essential component in the prognosis of a cancer. This paper aims to improve the prediction of survivability of urinary bladder cancer with a statistical and machine learning approach.**

***Index Terms*—UBC, incidence, machine learning, prognosis, morality.**

I. Introduction

Urinary bladder Cancer (UBC) is common around the world. Anytime about 3 million individuals have a past filled with UBC. The incidence of UBC differs over the world with most elevated rates in Western Europe and the United States. In any case, the percentage of UBC will increment in less created regions of the world. These progressions can be ascribed to worldwide changes in exposure to hazard factors for UBC and development and maturing of the whole world populace [1].

In excess of 12 million new instances of malignancy happen every year around the world. Out of those 5.4 million happen in developed nations and 6.7 million in developing nations [2, 3]. Urinary bladder cancer positions ninth in overall malignant growth incidence. It is the seventh most regular cancer in men and seventeenth in ladies [2]. Insights on the occurrence of bladder malignant growth are especially difficult to translate, due to evolving classification. Scientists are currently attempting to decide whether tests that distinguish genetic changes in bladder malignancy cells can help anticipate the survivability of the patient, which may influence treatment and several research approaches [3].

Advancement in medical research has huge impact on the survivability rate from cancer diseases because it leads new methods of treatment which are better than the methods available. Similar approach is put in the techniques of machine learning for the prediction of various outcomes of the disease. Different factors give different predictions results and different machine learning algorithm give different results, but the model which give almost always accurate prediction is considered to be a good model.

Massive amount of data is available online for almost every type of cancer on the SEER website (https://seer.cancer.gov). From the year 2008, out of 100 people that go through UBC, 77 have survived and these numbers are even lower for 1980-2000. This data is to be utilized in this paper for training different machine learning models and finding the best one out. Specific variables are chosen after feature selection to get better results.

This paper aims to improve the prediction of survivability for UBC patients by first classifying the survivability ranges and then using several machine learning algorithms to predict the survivability class of the patient.

II. Methods

A. *Data source*

The data for training the models was fetched from SEER database (1973-2015). SEER*Stat software, provided by the SEER website, was used to get specific variables required for the training. The training dataset consisted of about ~40 thousand data points after data cleaning.

B. *Feature selection*

The SEER database has huge amount of data with a handful of variables like age, sex, year of diagnosis, etc. Out of the variables (features), some are chosen to outperform the other. Choice of these features is an important part of the whole process as they will lead us to a good or bad predictive model. Machine learning algorithms work on a simple rule, if you insert trash in, you will get trash out. Sometimes the results from using subsets of the whole feature list gives far better results when compared to using whole features for the same algorithm. Machine learning algorithms need assistant when the feature list is huge, because the amount of unnecessary data (noise) could shift the results to a bad prediction and less accuracy. Therefore, it’s important to select only the important features (having greater classification importance) as a subset of full feature list. Moreover, feature selection is reducing the complexity and over-fitting, increasing the accuracy and speed of the model. SEER database has several unknown data fields due to the unrecorded data. This is mostly common in older data fields. These fields are considered as a problem for training the model because the results will lead to inaccurate predictions. Features having similar time ranges were kept, while the ones with unmatched time ranges were omitted.

Features from SEER database has to be picked wisely so our model can predict fast and accurate. So, to serve this purpose, sequential feature selection (SFS) approach is used to find the best combination of features that may lead to better results. Before applying forward SFS, those variables which gave very low variance are omitted. Some variables like age, sex are not passed in this process and are later added to the final subset, this is due to the obvious importance of these variables.

C. *Sequential feature selection*

Sequential feature selection is wrapper method for selecting important features (feature selection), where a subset is passed into the algorithm and based on the results, a features is added or removed from the final subset. These type of methods are very computationally expensive as compared to filter methods like Person's correlation, LDA, Chi-Square, etc. The reason being, passing list of several features from the algorithm several times to get the best subset. This could be in thousands or hundreds of thousands for huge set feature list.

There are mainly two types of SFS that are; forward SFS and backward SFS. Forward SFS uses an iterative approach that works simply by having no feature at the beginning, then in every iteration, the best feature among all features is added into the final subset. These iterations keeps on adding features until further addition of feature doesn't improve the model accuracy. Backward SFS is similar to forward SFS expect in this method, final subset has all the features in it and in every iteration the least significant features is removed from the final subset to increase the accuracy. These iterations keeps on removing the feature until further removing of feature doesn't improve the accuracy. This paper uses forward SFS to compute the best subset of features. Massive amount of features list reduced to obtain only 12 features in the final subset.

I. Age groups (5 years gap)
II. Sex
III. Year of diagnosis
IV. Marital status
V. Cancer grade
VI. Tumor size
VII. Lymph nodes
VIII. Total number of insitu tumors
IX. Histologic type ICD-O-3
X. Primary site
XI. Derived AJCC
XII. Regional positive nodes

D. *Classification of survivability*

SEER database provide us with a variable 'survival months' which is the amount of months individual survived after the diagnosis. This variable is used as the prediction outcome after splitting it up in 5 different classes.

TABLE I
SURVIVABILITY CLASSES

5 different classes		
Class #	Comment	Survivability year range
0	Death due to UBC	-
1	Surviving more than 30 months	2.5 years +
2	Surviving more than 60 months	5 years +
3	Surviving more than 90 months	7.5 years +
4	Surviving more than 120 months	10 years +

E. *Prediction models*

The advancement in machine learning techniques has brought various algorithms which work best on several different datasets. To find the best model that solves the problem in the most efficient way is still a very extensive task. In this paper, four classifiers are used to determine the survivability class.

a. Naive Bayes

In naïve Bayes classifier the features are assumed to be independent to one other, which makes it the simplest classifier (naïve). The assumption of independence of features is poor assumption for a classifier but still NB competes well with other complex algorithms like SVM, KNN, etc [4].

$$P(\boldsymbol{X}|C) = \prod\nolimits_{i=1}^{n} P(X_i|C)$$

Where, $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ *feature vector*
C = *class*

b. K-nearest neighbors

In KNN an object is classified by majority weight by its neighbors. The class of an object is determined by the class of its nearest k neighbors. To find the nearest neighbor x_{NN} of a given query point $q \in R^d$ in the database $D \subset R^d$ is defined as [5]

$$x_{NN} = \{x` \in D | \forall x \in D, x \neq x`: dist(x`, q) \leq dist(x, q)\}$$

c. Decision trees

A decision tree algorithm uses a tree like structure to find the most probable class of a set inputs. This algorithm is based on condition control statements, one condition leads to a set of another conditions to be executed. An example of simple decision tree is shown in Fig. 1 [6].

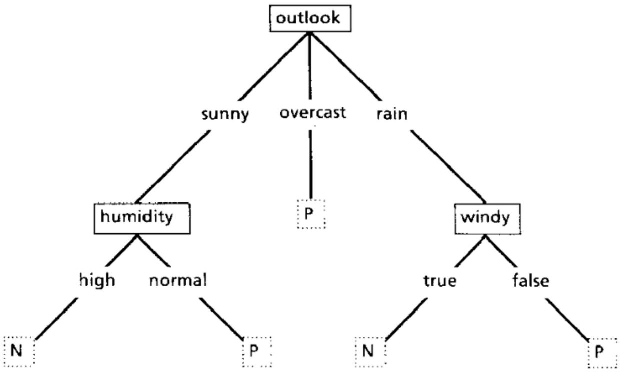

Fig. 1. A simple decision tree example

d. Support vector machine

SVM works on the phenomenon of supervised learning. It is a discriminative classifier, which is defined by a hyper-plane separating the classes. A labeled training dataset is given as an input to the classifier, while an optimal hyper-plane categorizing the classes of new examples is provided as an output. In two dimensional space, the hyper-plane is a line dividing the two classes (binary). Similarly, in three dimensional space, a two dimensional plane divides the three classes categorizing them for further predictions of test dataset [7].

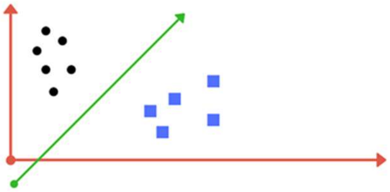

Fig. 2. A simple support vector machine example

F. Performance metrics

The evaluation of the machine learning model is done based on several metrics and to choose certain metrics is very important as it influence's the performance of the machine learning model. These metrics helps to compare and measure different machine learning models, which further leads to selecting the best model for a certain dataset. Before jumping into the metrics, a confusion matrix is drawn from which various metrics are calculated. Confusion matrix is a simple 2x2 matrix having cells named TP (true positive), TN (true negative), FP (false positive), and FN (false negative). Here positive and negative refers to the actual survivability, while true and positive refers to the predicted survivability. From these 4 quantities several other metrics are calculated from these formulas:-

a. Accuracy

Accuracy is the measurement of all correct predictions made in all types of predictions made. This metric is useless when the classes are not approximately evenly distributed.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

b. Precision

Precision is the measurement that gives information about what percentage of patients that the model predicted of surviving actual survived.

$$Precision = \frac{TP}{TP + FP}$$

c. Sensitivity

Sensitivity is the proportion of the patients that were predicted to survive in certain class, were actually able to survive that class.

$$Sensitivity = \frac{TP}{TP + FN}$$

d. Specificity

Specificity is the opposite of the sensitivity. Here the output is the proportion of the patient's survivability class that was predicted not surviving end up actually not surviving.

$$Specificity = \frac{TN}{TN + FP}$$

e. F1 Score

F1 Score is the harmonic mean of precision and sensitivity. This one metric gives the result of two metrics combined.

$$F1\ Score = 2 \times \frac{Precision \times Sensitivity}{Precision + \mathrm{Sensitivity}}$$

K-cross validation is used to avoid over-fitting of the model so it performs well when out in practice. K-cross validation workings by the splitting dataset into k subsets (S_1, S_2, S_3,…, S_k) out of which one subset is chosen as test dataset and rest are used for training the model. Similarly in the next iteration another subset (fold) is chosen as a test set and rest as training set. This keeps on repeating until all the subsets (folds) are used as test sets. The final accuracy is obtained by taking average of the accuracies of the iterations.

$$Final\ accuracy\ (A) = \frac{\sum_{i=k}^{k} A_i}{k}$$

A_i = Accuracy of the i^{th} iteration
k = Number of folds (subsets)

III. RESULTS

The metrics used for the evaluation of machine learning algorithms plays a vital role in equal comparison of different classifiers. The metrics chosen for evaluation here are accuracy, precision, sensitivity, and f1 score. Four classifiers were used to train a model and then tested upon the mentioned metric. The results are shown in Table 2. The results show us that SVM (support vector machine) have highest metric values (accuracy, precision, sensitive, and f1 score) as compared to other classifiers (Naïve Bayes, KNN, and Decision trees).

TABLE II
RESULTS

Naive Bayes		
Metric	Metric value	Standard deviation (Kfold)
Accuracy	79.1821%	4.09%
Precision	79.3662%	4.32%
Sensitivity	79.1821%	4.09%
F1 score	79.2741%	4.20%
K-Nearest neighbors		
Metric	Metric value	Standard deviation (Kfold)
Accuracy	77.6060%	5.19%
Precision	77.6980%	6.01%
Sensitivity	77.6060%	5.19%
F1 score	77.6520%	5.57%
Decision trees		
Metric	Metric value	Standard deviation (Kfold)
Accuracy	81.4348%	5.83%
Precision	81.5747%	6.18%
Sensitivity	81.4348%	5.83%
F1 score	81.5047%	6.0%
Support vector machine		
Metric	Metric value	Standard deviation (Kfold)
Accuracy	83.2214%	4.53%
Precision	84.8591%	4.04%
Sensitivity	83.2214%	4.52%
F1 score	84.0323%	4.27%

The maximum accuracy is 83.22% (4.53% standard deviation) which is achieved by SVM and Decision trees stood second place with 81.43% (5.83% standard deviation). The least accuracy was shown by KNN. Machine learning techniques have its own limits when it comes to using them in the real world. These algorithms use a statistical approach in decision making which could be different from the actual reality. The use of this paper should help in assisting an individual in the final decision making.

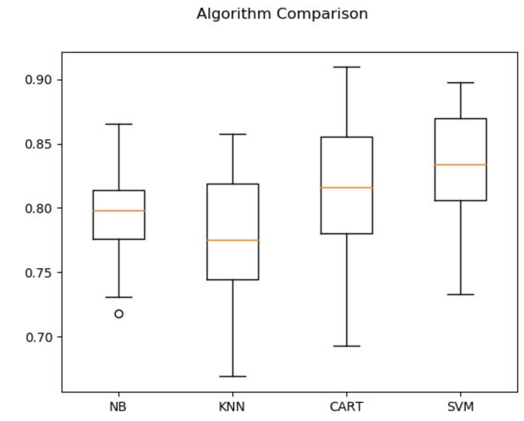

Fig. 3. Results shown in a candlestick chart

A GitHub repository tool is created based on this paper (https://github.com/iamkhalidbashir/UBCpredict), which will further lead other researchers to contribute in improving the efficiency of these results mentioned in this paper.

IV. CONCLUSIONS

Machine learning models are only efficient if the dataset used for its training is well produced, meaning that a noise in the input training data set will lead the model to give noisy (bad) results. The dataset provided by SEER website was not used as it is for the model generation, but were first cleaned to remove any data points that may invalidate the output. SFS (sequential feature select) techniques were implemented to select he best features form the list of all features provided with the SEER dataset. This further made our model more efficient in terms of accuracy and speed. Different evaluation metrics were considered for the comparison of different machine learning classifiers, so a good model could be produced as it is important for an application in the medical field. After comparison the results were examined giving the best output by SVM (support vector machine) which is 83.22%. Decision trees stood second at 81.43%. A tool was finally generated and uploaded to the GitHub repository, so the medical and machine learning community could get assist in predicting the urinary bladder cancer in an individual or to further improve the results mentioned in this paper. A caution is must before considering the prediction of this tool as a final outcome as it’s a matter of life and death and these tools could be biased in their results for some cases.

REFERENCES

Papers:

[1] Garcia M, Jemal A, Ward EM, Center MM, Hao Y, Siegel RL, Thun MJ (2007) Global cancer facts and figs 2007. American Cancer Society, Atlanta.

[2] Ferlay J, Bray F, Pisani P, Parkin DM (2004) GLOBOCAN 2002: cancer incidence, mortality and prevalence worldwide. IARC CancerBase No. 5.version 2.0. IARC Press, Lyon.

[3] Advanced Bladder (ABC) Meta-analysis Collaboration. Adjuvant chemotherapy in invasive bladder cancer: A systematic review and meta-analysis of individual patient data. European Urology. 2005;48:189-201.

[4] Rish, Irina. "An empirical study of the naive Bayes classifier." *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. No. 22. New York: IBM, 2001.

[5] Hinneburg, Alexander, Charu C. Aggarwal, and Daniel A. Keim. "What is the nearest neighbor in high dimensional spaces?." In *26th Internat. Conference on Very Large Databases*, pp. 1-2. 2000.

Books:

[6] Quinlan, J.R., 1986. Induction of decision trees. *Machine learning*, *1*(1), pp.81-106.

[7] Cristianini, N. and Shawe-Taylor, J., 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.