Analysis of Prostate Cancer Dataset to find Meaningful Bio-markers Using Machine Learning Techniques

Parth A. Shukla & Dipesh S. Patel

Abstract— **Prostate cancer (PC) is the second most regular type of disease in men around the world. Bio-markers have developed as basic devices for treatment and appraisal since the changeability of ailment conduct, the expense and decent variety of medicines, and the related weakness of personal satisfaction have given rise to a requirement for a customized methodology. High-throughput innovation stages in proteomics and genomics have quickened the improvement of bio-markers. This project presents an advanced approach towards finding best bio-markers for PC by harnessing the computational power of machine learning.**

Index Terms— **Machine Learning, Naive Bayes, K-nearest neighbours, Support Vector Machine (SVM), Feature Selection, Principal Component Analysis, Over Sampling**

I. INTRODUCTION

Prostate cancer is the development of cancer in the prostate, a gland in the male reproductive system. Most prostate cancers are slow growing; however, some grow relatively quickly. The cancer cells may spread from the prostate to other areas of the body, particularly the bones and lymph nodes. It may initially cause no symptoms. In later stages, it can lead to difficulty urinating, blood in the urine or pain in the pelvis, back, or when urinating. With current advancement in biotechnology and medical science, machine learning has played a major role in engendering efficient and quick methods which are helpful in classifying the progress of malignancies in patients. We take the Prostate cancer dataset from CBioPortal which contains around approximately 60,000 features with 494 samples. As cancer data has a vast amount of gene expressions and feature imbalances, we consider feature selection as the initial phase of our implementation. Based on the best features, to eliminate the problem with imbalanced data, we perform over sampling and then perform classification based on which we evaluate our results.

This report is organized as follows. In section II we discuss various methods and algorithmic details of the models that were used as part of this project. Section III explains the systematic approach towards the problem. Section IV depicts the results. Section V presents the analysis on the result based on the the approach and we conclude the report in Section VI.

II. METHODS

This section focuses on the algorithms used in various phases of our implementation. These methods are core machine learning approaches which can enhance any classification problem.

A. Feature Selection

Feature Selection is a prime paradigm which is used by a majority of statisticians and computer scientists to select a subset of relevant features (variables, predictors) which can be further used for model construction. This approach is mainly used because of the following reasons:

- There is a need to simplify a model for easy interpretation by researchers
- To reduce computational complexity while training the model
- To avoid the curse of dimensionality: A phenomena that arise when analyzing and organizing data in high-dimensional spaces (often with hundreds or thousands of dimensions) that do not occur in low-dimensional settings such as the three-dimensional physical space of everyday experience
- To enhance generalization of features to reduce over-fitting.

For this project we take Random Forest and Chi-squared feature selection methods into consideration.

1) Random Forest: Random forest (RF) is a supervised learning algorithm which creates an ensemble of decision trees based on which data is passed in form of the random vectors and results are derived [1]. Each tree in the forest is given a random vector of data features which is sampled independently. Based on the samples given to the decision trees, each tree contributes to the decision rule to classify the sample [1]. The forest chooses the class with the majority of the votes [1]. The motivation behind using technique is:

1) RF handles the missing values and maintains the accuracy of the data [1].
2) This algorithm never over-fits the model.
3) It can easily handle large datasets with high dimensions [1].

A simplified pseudocode for random forests is shown below:

- Assume number of cases in the training set to be N. Then the sample of these N cases are fed into a random vectors with replacement
- If there are m input features, where m¡M is specified such that at each node, m features are selected at random out of M. The best split of these m is used to split the node. The value of m is held constant while we grow the forest.
- Each tree is grown to the largest extent possible and there is no pruning.
- Based on the classification, majority of votes are collected and at the end the class that got highest vote is returned.

2) Chi-Squared: Essentially chi-squared methods measures the degree of independence of a feature to the classes [2]. A set of features is passed in this method based on which the features will be ranked based on the index and we can select *i* number of features from total *n* features. [2]. Mathematical equation is as shown below.

$$chi^2(y,x) = \frac{n(ad-cb)^2}{(a+c)(b+d)(a+b)(c+d)} \quad (1)$$

where,

- a is the number of times feature x an class y co-occur
- b in the number of times x occurs without y
- c in the number of times x occurs without y
- d in the number of times neither x and y occurs
- n is the total number of samples

The motivation behind using technique is:

1) Robustness with respect to distribution of the data
2) Easy to compute
3) Flexibility in handling data high dimension data

B. Classification

In machine learning, classification is a technique to predict the class label of a unknown sample based on a trained model. For example a dataset consisting data related to some few features of and cat and dog based on which, a classification model X is trained. After training, the trained model, M(X) should classify any unknown sample x to the correct category. i.e. either a dog or a cat.

Classifier is first trained with set of samples S. After the model is trained, it is tested with unknown samples S'. For this purpose generally we perform m-fold cross validation where datasets is divided into m equal parts

$$S_1, S_2, ..., S_m$$

out of this all folds m-1 folds are taken for training and the remaining fold is taken up for testing. This is performed for all m folds leading to m results which are averaged out leading to final result of the classifier.

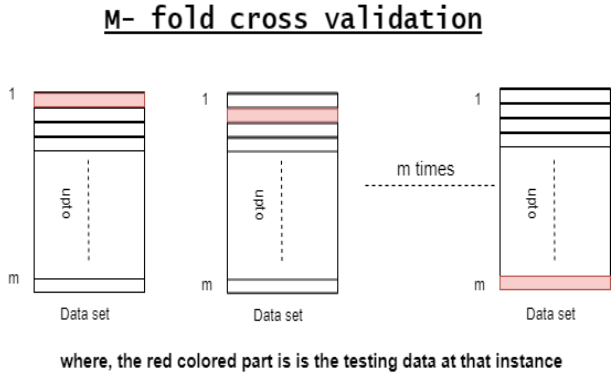

Fig. 1. M-fold cross validation

There are many types of classification algorithms working on different mathematical paradigms but they can be broadly classified in two

types. One is the supervised learning algorithms and other are the unsupervised ones. For this project, based on the nature of dataset, we consider supervised classification algorithms.

1) Support Vector Machines: Support Vector Machines (SVM) is one of the most widely used method for classification. It is very suitable for small datasets with large number of features. This algorithm takes data from the lower dimensional *Y* space and derives a linear function in a higher dimensional space *Y'*. Classification is performed on higher dimensional space [3]. This method makes use of support vectors to derive the classifier.

Aim is to maximize the distance between the nearest point of either classes and the hyper-plane [3]. This distance is called as margin.

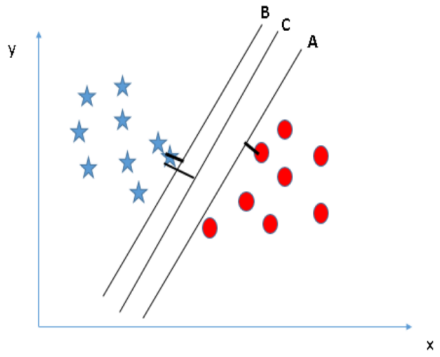

Fig. 2. Hyper plane selection in SVM

Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification. Not all classification problems can be approached with a linear function with a separating hyper-plane. There are cases when there are multiple classes and classification boundary on higher dimensions in non-linear. In such cases we perform technique called the kernel trick. These are functions which takes low dimensional input space and transform it to a higher dimensional space i.e. it converts not separable problem to separable problem, these functions are called kernels. It is mostly useful in non-linear separation problem. For this dataset we take Radial Bases Function (RBF) into consideration. We take two tuning parameters into consideration C and γ.

The γ, parameter can be thought of as the spread of the kernel and therefore the decision region. With low value of γ the decision region is very broad. But with a high value of γ , the decision boundary is high, which creates islands of decision-boundaries around data points. On high increase in the value, islands are formed which leads to over-fitting of data and leading to poor classification prediction capability.

The C parameter trades off correct classification of training examples against maximization of the decision functions margin. For larger values of C, a smaller margin will be accepted if the decision function is better at classifying all training points correctly. A lower C will encourage a larger margin, therefore a simpler decision function, at the cost of training accuracy. In other words C behaves as a regularization parameter in the SVM.

2) Naive Bayes: This approach focuses on quantifying trade offs between various classification decisions. Furthermore, the approach focuses on conditional probabilities and the cost associated with those decisions [4]. Let ω denote the classes of the data points. We have 'c' classes

$$\omega_1, \omega_2, ..., \omega_c$$

For this probability distribution, posterior probabilities are calculated such that P(ω_1) + P(ω_2) + ... + P(ω_c) = 1. All thus posterior probabilities are fed to a conditional probability based decision rule [4]. Based on the class conditional probability we can formulate,

$$P(\omega_j/x) = \frac{p(x/\omega_j)P(\omega_j)}{p(x)} \tag{2}$$

where,

$$p(x) = \sum_{j=1}^{c} p(x/\omega_j)P(\omega_j) \tag{3}$$

Based on above mathematical explanation, if we have

$$x_1, x_2, ..., x_d$$

independent data-points, we have have to predict the probability of belonging to a particular classes that is p(x/ω_j) which can be calculated as:

$$p(x/\omega_j) = \prod_{i=1}^{d} p(x_i/\omega_j); j = 1, 2, ..c \tag{4}$$

Based on this, the point gets classified into class which gives highest probability result [4].

3) K Nearest Neighbours: K-Nearest Neighbors is a non-parametric, lazy learning algorithm. Its purpose is to use a database or dataset in which the data points are separated into several classes to predict the classification of a new sample point [4]. When we say a technique is non-parametric, it means that it does not make any assumptions on the underlying data distribution. K-Nearest Neighbors is considered a lazy learning algorithm because it does not use the training data points to make any generalizations. In this algorithm a positive integer 'k' is specified, along with a new sample [4]. We then select the k entries in our database which are closest to the new sample after the distance is calculated [4]. The distance formula used in our approach is the Euclidean distance. We find the most common classification of these entries. This is the classification we give to the new sample.

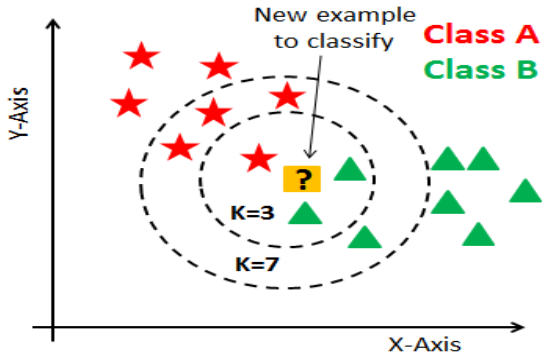

Fig. 3. K-Nearest Neighbor Working

As we can see from Fig. 3 after the distance is calculated the number of surrounding neighbors are checked. As we can see for K=3, there are 2 data points of class B closest to the new example to classify. So the new example would be classified in class B for K=3. If we consider K=7, the new example would be classified in class A as there are more data point of class A closer than class B.

C. Over-Sampling

In actual datasets, usually there is a case where data is unbalanced in terms of the representation of the data sample population. If such data is fed to the model for training purpose, the predictions won't be accurate. This problem is called class imbalance problem which can be solved by over sampling of data.

The main motivation behind the need to pre-process imbalanced data before we feed them into a classifier is that typically classifiers are more sensitive to detecting the majority class and less sensitive to the minority class. Thus, if we don't take care of the issue, the classification output will be biased, in many cases resulting in always predicting the majority class. Here we use two flavours of oversampling algorithms which are described ahead.

1) Naive Random over-sampling: In this approach, the algorithm calculates the highest majority count of classes and based on that value the algorithm selects a data sample from least majority and generate new samples by randomly sampling with replacement the current available samples.As a result, the majority class does not take over the other classes during the training process. Consequently, all classes are represented by the decision function. Example of this algorithm can be seen in Fig. 7.

2) Synthetic Minority Oversampling Technique (SMOTE): This algorithm takes K-nearest neighbour into account [5]. In SMOTE, on considering a sample x_i, a new sample x_n will be generated based on k-nearest neighbour and a new point x_z is selected and sample is generated based on the math shown below:

$$x_n = x_i + \lambda \times (x_z - x_i) \tag{5}$$

where λ is a random number in range [0,1]. This interpolation will create a sample on the line between x_i and x_z. Example of this algorithm can be seen in Fig. 8.

III. APPROACH

Initially we take the two .csv files obtained form CBioPoral where one file contains 494 samples and other file contains all the 60494 gene expressions which are essentially features for our project. At first we want to remove noise from the data. On observing feature set, we get many features that had all the data as value '0', which was irrelevant feature and there were few thousand of those. So first, a script was executed to remove such noisy

features. After that we perform inner join between those two datasets based on the patient ID from gene expression dataset and create a combined data frame to make data easily accessible to code.

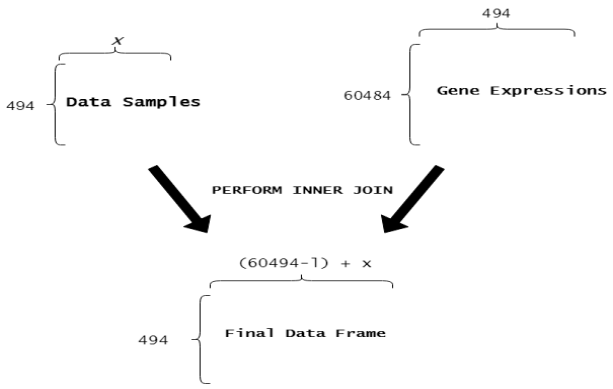

Fig. 4. Combined Data frame

Here you can see, the data samples had shape of $[494 \times x]$ and Gene expression had shape of $[60484 \times 494]$ which after removing l number of completely 0 containing rows and performing inner join gives final shape of $[494 \times (60494 - l + x)]$. Here x is the number of class labels.

After the processing on the data and merger is complete, we perform feature selection based on which we want to get best features for our model. For feature selection we take Random Forest and chi-sqaured algorithms. After feature selection, we need to balance out the classes. Here we have taken LATERALITY as out class set which consist of three classes; $left$, $right$ and $bilateral$ out of which the population of $bilateral$ dominates the rest two. For oversampling we follow two approached, one is the Naive Random Oversampling method and other is the SMOTE. After Oversampling, we classify our data and evaluate the results using KNN, Naive Bayes and SVM-RBF algorithms.

The histogram shows the state of data before and after over sampling was performed.

IV. RESULTS

A. Over Sampling Results

As shown in Fig 6 , 7 and 8 (pg.6) we have the PCA visualized plots for actual data and data after oversampling. Here we can see that there is not much difference between the actual data plot and the data plot after Naive Random oversampling as it just duplicates the existing the actual existing data to satisfy the sample representation of each class.

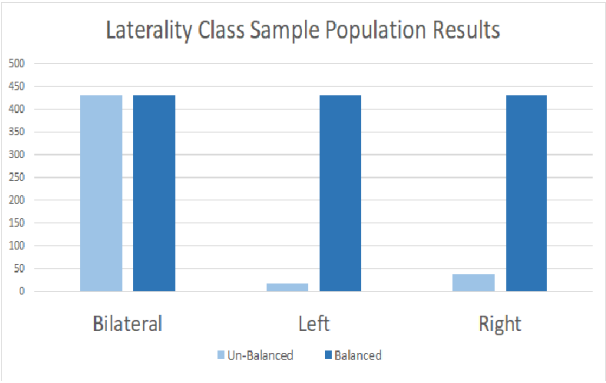

Fig. 5. Class Sample Population Analysis

Initial state of dataset was ⟨bilateral, left, right ⟩ = ⟨ 430, 18, 38 ⟩ which got transformed to ⟨ 430, 430, 430⟩ which reduced the bias of bilateral class on the entire classification and training models.

B. Feature Selection Accuracy Results

Below the feature and classification results. Note, after the feature selection, oversampling was performed which was followed with classification and then prediction. Here in the tables where there are comma separated values besides the accuracy score are the number of features selected by the feature selection algorithm for that iteration.

We have four accuracy reports out of which I and II (pg.7) belongs to the Random Over Sampling category and III (pg.7) and IV (pg. 7) belong to the SMOTE category.

chi^2 features	Naive Bayes	SVM-RBF	KNN ($k = 22$)
10	42.945736%	96.2453%	70.3875%
20	42.79069%	96.465%	74.806%
30	48.75968%	100.0%	74.806%
40	52.63565%	97.456%	74.6511%
50	58.372093%	100.0%	74.4961%
60	61.317829%	99.4568%	74.57364%

TABLE I
ACCURACY REPORT - I

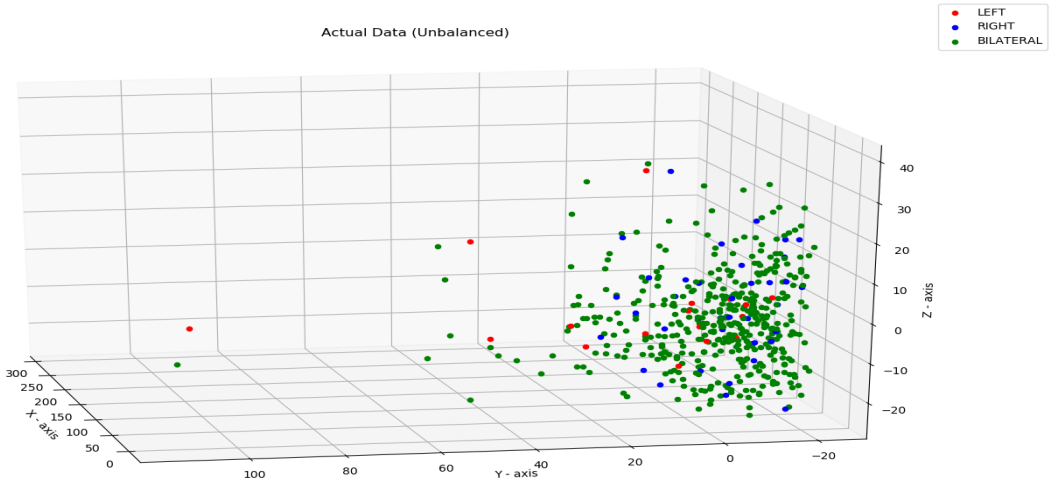

Fig. 6. PCA on Data

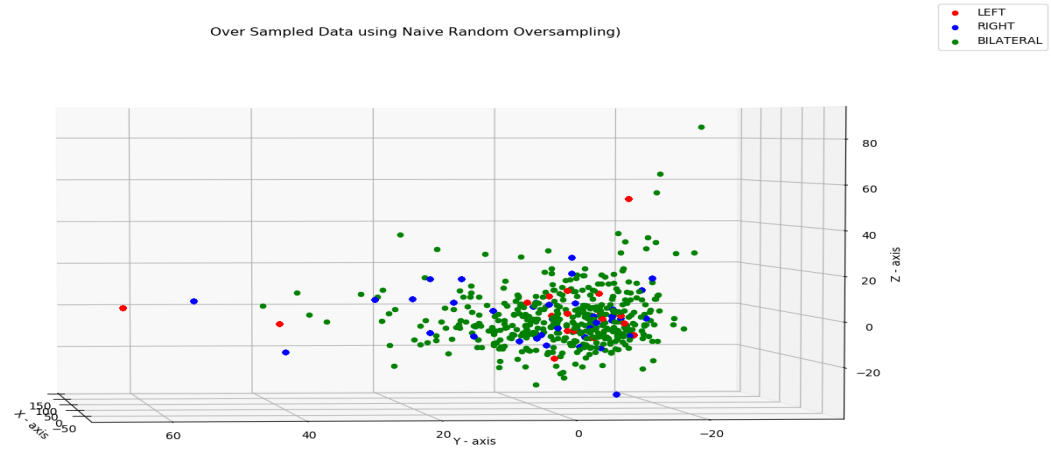

Fig. 7. PCA on data with Random Over Sampling

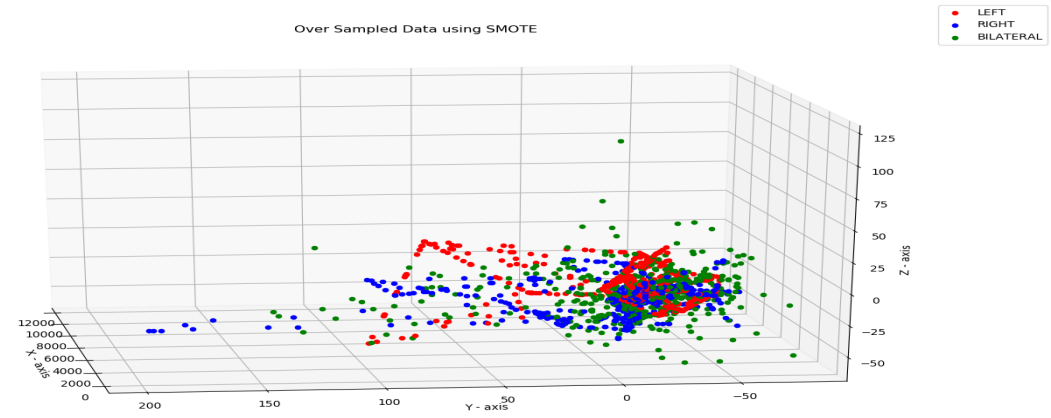

Fig. 8. PCA on data with SMOTE

RF estimators	Naive Bayes	SVM-RBF	KNN ($k = 22$)
1	52.868% , 31	99.922% , 32	75.271% , 36
2	58.1395% , 59	100.0%, 58	69.069%, 67
3	67.519% , 97	100.0% , 97	72.015%, 103
4	71.007% , 128	100.0% , 127	71.472% , 128
5	73.488% , 162	100.0% , 167	70.155% , 156

TABLE II

ACCURACY REPORT - II

chi^2 features	Naive Bayes	SVM-RBF	KNN ($k = 22$)
10	49.224%	86.302%	70.775%
20	47.054%	83.062%,	76.124%
30	52.248%	90.3547%	74.263%
40	58.449%	87.256%	75.038%
50	64.341%	89.356%	74.728%
60	68.837%	87.856%	74.341%

TABLE III

ACCURACY REPORT - III

RF estimators	Naive Bayes	SVM-RBF	KNN ($k = 22$)
1	65.116% , 36	88.062% , 32	71.550% , 30
2	67.364% , 58	88.062%, 63	73.875%, 64
3	73.1% , 91	90.387% , 91	72.403%, 98
4	80.38% , 123	64.031% , 124	72.248% , 132
5	79.069% , 150	59.844% , 157	74.883% , 164

TABLE IV

ACCURACY REPORT - IV

V. DISCUSSION

A. Accuracy analysis

- From a cursory reading on the accuracy reports it's clear that SVM-RBF performs the best in comparison to KNN and Naive Bayes. The reason why SVM performs best is because it makes data separable on higher dimensions and in come cases data turns out to be perfectly separated. Essentially SVM-RBF can perform better than other SVM kernels like linear and polynomial as γ and C can tune SVM to form uneven decision boundaries on higher dimensions.
- Here we have 484 samples in total for for KNN we take value of $k = \sqrt{484} \approx 22$. KNN performs good but not that efficiently as data is actually cluttered which can lead to poor decision in terms of distance based classification. So for this case, KNN is not applicable.
- Talking about Naive Bayes, it essentially fails to give better results as data is essentially possessing continuous nature. In this case, computing the probabilities by the traditional method of frequency counts is not possible which leads to poor probability distribution leading to poor conditional probability results ultimately leading to poor results.

B. Feature Selection Algorithms Analysis

- Here we take two algorithms into consideration, Random Forest and Chi-Squared which follows filter based approach. Given the number of features ($\approx$ 60K), wrapper approached like Recursive Backward Selection and Forward selection takes much more longer time to execute.
- For Random Forest, we specify the number of estimators which is shown in the first column in Accuracy report II and IV. Number of estimators represents the number of trees in the forest based on which features are selected. And for chi-square we have to explicitly define number of features we want from the feature set.
- Chi-Squared method is definitely computationally lighter in comparison with Random Forest, but here we are not aware of how many number of features exactly we want for classification. This becomes a classic two variable optimization problem where accuracy and number of features are inversely proportional and we need a optimal pragmatic solution at end.
- Considering a practical scenario , biologists want few 50-70 gene expressions (features) based on which they can work efficiently. This constraint is satisfied by Random Forest algorithm, plus it will never over fit the dataset which makes it better choice for feature selection. Based on this fact and our accuracy reports, we present best 32 gene expressions which is 0.05% of total gene expression dataset which gave best accuracy (pg. 8).

Name of Feature	GINI importance score
ENSG00000212144.1	0.02153239206212782
ENSG00000123572.15	0.037353965976634554
ENSG00000223595.1	0.04103127912863271
ENSG00000226669.2	0.09561577679852427
ENSG00000271271.4	0.04018836677291596
ENSG00000269792.1	0.019439931639800827
ENSG00000188283.10	0.07307399227275158
ENSG00000264145.2	0.0396255458715869
ENSG00000198625.11	0.02098632931575622
ENSG00000252677.1	0.058984530475554874
ENSG00000252918.1	0.040557067385511576
ENSG00000279921.1	0.026979426375274455
ENSG00000273124.1	0.01089490674318508
ENSG00000261717.4	0.021426902899876286
ENSG00000184361.11	0.01089490674318508
ENSG00000272865.1	0.02163280162366703
ENSG00000005102.11	0.04110856208140239
ENSG00000120875.7	0.010490392424659758
ENSG00000197496.5	0.03486370157819224
ENSG00000236859.5	0.058061970093816105
ENSG00000274211.3	0.021126467766023595
ENSG00000281685.1	0.021232129691702403
ENSG00000207242.1	0.02097899894089847
ENSG00000277258.3	0.01743185078909612
ENSG00000203999.7	0.041439632470115666
ENSG00000168748.12	0.014526542324246772
ENSG00000233205.1	0.021084856683436244
ENSG00000121742.14	0.039824671146011334
ENSG00000224843.5	0.02088849095064842
ENSG00000266849.1	0.010635265516971915
ENSG00000267121.4	0.0043054571226080526
ENSG00000178081.11	0.021274828583307523

TABLE V

BEST FEATURES BASED ON ACCURACY REPORT

C. Oversampling Algorithms

- Based on the discussions, it is clear that SMOTE is better choice than Naive Random over-sampler. SMOTE takes the interpolation process and generates synthetic data points which essentially better than duplicating the original points to match the majority. Illustrations are on pg. 6.
- So, SMOTE approach is practical and useful for developing synthetic data which is essentially near to the unknown sample probability as it uses KNN to locate the best position for a new synthetic data point for value of k.

VI. CONCLUSION

In this report we dealt with a multi-class , data imbalanced high dimensional data of Prostate cancer and we see that there are lot of challenges while dealing with real time data like data-cleaning, prepossessing and merging before making actual use of data. We perform machine learning approaches to the classification problem using multiple subsequent approaches like Feature Selection, dimensional reduction, oversampling and classification.

Based on the results and discussion we conclude that for this problem, we can use RBF kernel for classification, Random Forest for feature selection and SMOTE for managing class imbalance problem. There are some cases where we can make feature selection more efficient by applying wrapper based feature selection method on actual obtained features from Random Forest to cut down features to more convenient number.

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our course instructor Dr. Luis Rueda and our research supervisor Dr. Saeed Samet for their continuous support in our project, patience, motivation, and immense knowledge. Their guidance helped us throughout our time of research and implementation of this project. We could not have imagined having better mentors for our study.

REFERENCES

[1] L. Breiman, "Random Forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[2] X. Jin, A. Xu, R. Bie, and P. Guo, "Machine learning techniques and chi-square feature selection for cancer classification using SAGE gene expression profiles." Springer-Verlag, Apr. 2006, pp. 106–115. [Online]. Available: http://dl.acm.org/citation.cfm?id=2097024.2097039

[3] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995. [Online]. Available: https://doi.org/10.1023/A:1022627411411

[4] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, Sep. 1999. [Online]. Available: http://doi.acm.org/10.1145/331499.331504

[5] S. W. Purnami and R. K. Trapsilasiwi, "SMOTE-Least Square Support Vector Machine for Classification of Multiclass Imbalanced Data," in *Proceedings of the 9th International Conference on Machine Learning and Computing*, ser. ICMLC 2017. New York, NY, USA: ACM, 2017, pp. 107–111. [Online]. Available: http://doi.acm.org/10.1145/3055635.3056581