A Machine Learning Approach for Finding Meaningful Genes in Prostate Cancer

Aman K. Patel[1] and Aditya Rao[2],

[1]School of Computer Science, University of Windsor, Windsor, ON N9B3P4 Canada
[2]School of Computer Science, University of Windsor, Windsor, ON N9B3P4 Canada

Corresponding author: Aman K. Patel (e-mail: patel1wq@uwindsor.ca).

Corresponding author: Aditya Rao (e-mail: rao113@uwindsor.ca).

ABSTRACT Prostate cancer is considered as the most prevalent cancer and is one of the major causes of death all over the world. It affects the prostate gland in male reproductive system. Genetic information of prostate cancer patients may provide us with some useful information about a cell. In this paper, we use a comprehensive dataset that includes the gene information and clinical information of 494 patients. We tried training the model using several different standard classifiers and choose the one which gave us the high performance. A combination of feature selection methods was applied to identify the genes that can predict the Gleason score of a patient. The result shows that the Random Forest was able to identify the Gleason score with high performance measure.

INDEX TERMS Gleason score, Naive Bayes, SVM, LDA, K Nearest Neighbor, Random Forest, Filter feature selection, Wrapper feature selection

I. INTRODUCTION

Prostate Cancer is a cancer that occurs in the prostate. The prostate is a small walnut shaped gland in males that produces the seminal fluid that nourishes and transport sperm. About 248,530 new cases of prostate cancer, About 34,130 deaths from prostate cancer[1]. Prostate cancer can be graded using Gleason score. It helps practitioners to understand the behavior of a cell at microscopic level. This grading system can be used to choose appropriate treatment options. Typical Gleason Scores range from 6-10. The higher the Gleason Score, the more likely that the cancer will grow and spread quickly[2]. We developed a methodology using various machine learning techniques to identify meaningful gene biomarkers necessary for predicting Gleason scores. The motivation behind our study was the fact that the tradition research and work done was on detection of benign and malignant tumors while doing the initial research. Identification for the useful features requires accurate and informative gene expression data of individual patients.

To give a slight sneak of the complexity related to the dataset interpretation, it consists of thousands of gene expressions which consists of imbalances that needed to be balanced using sampling methods. Also, we knew that amongst those thousands of genes expression there are some features which are significant and relevant to the problem addressed in the machine learning model. Since Gleason Score value ranges from 6 to 10, this problem becomes a multiple class problem. We use one vs rest classification technique on the given clinical data joined with the corresponding gene data.

II. METHODOLOGY

The entire process was divided into several parts:

A. Understanding the data
B. Preprocessing the data
C. Feature Selection
D. Classification

A. Understanding the data

We used the publicly accessible dataset available at cBioPortal Prostate Adenocarcinoma. The dataset was provided to us in two parts, one was the clinical data which contains the clinical information of a patients (T-Score, Gleason score, life expectancy and many more), second data set contains the quantitative information of 60489 genes for the same patients. Since we wanted to find the genes related to Gleason score prediction we choose Gleason score from clinical dataset as the label and trained the classifier to accurately predict the Gleason score using the Genomic data. Gleason Score is the grading system used to determine the aggressiveness of prostate cancer.

After studying the given data, we noticed there were a few insignificant features. We were also able to identify the class imbalance which can be visualized in Fig. 1

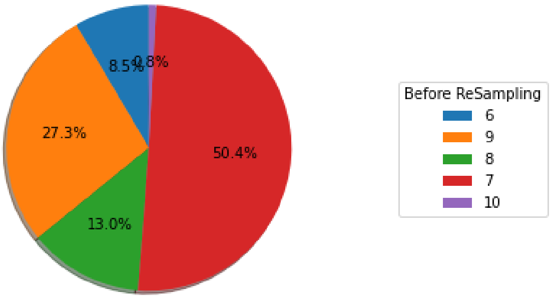


Fig. 1 Distribution of Gleason Score

B. Preprocessing the Data

Removing those features which had 0 values (NaN values) for all patients.

There are 3018 features (genes) whose values are 0 (NaN) for all the patients. We decided to remove those features since they will not contribute towards decision making.

Class Imbalance

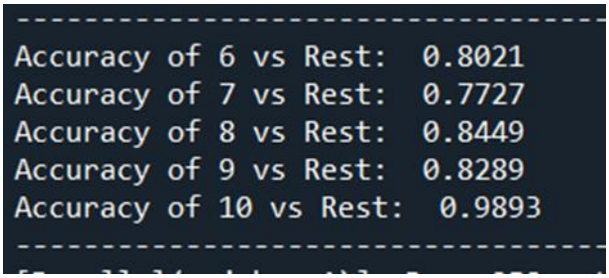
Number of examples that belong to each class is referred to as class distribution. An imbalanced classification problem is an example of a classification problem where the distribution of samples across the known labels is biased or skewed. The distribution can vary from a slight bias to a serve imbalance where there is one example in the minority class for hundreds, thousands, or millions of samples in the majority label or labels. Our model uses one-versus-rest scheme to tackle this multiclass classification problem. Fig. 1 shows that out data is imbalanced and thus there is a need to balance the data.

Why is there a need for class balancing?

Predictive models can be constraint to imbalances, since the models were designed with the hypothesis of an equal number of samples for each class. This results in poor performance of the model when predicting, especially the minority labels. The problem becomes more sensitive when the minority classes become more important and therefore the classification errors of the minority class than the majority class. Hence, it is very important to balance the classes since we don't have enough samples to fit the model on for the minority class, we are unable to predict the minority class properly which is going to be discriminatory towards the minority classes because of the naivety of the model which focuses on learning the characteristics of the abundant observations only.

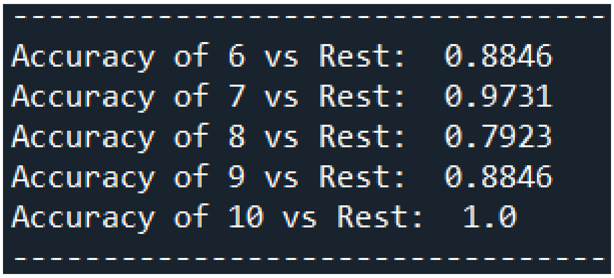
Two resampling techniques are generally used to tackle this problem: 1. generating synthetic data instances of minority class, which is called oversampling 2. Deleting the data instances of the majority classes, which is called under sampling. Under-sampling of the majority class has been proposed as a good means of increasing the sensitivity of a classifier to

the minority class. This paper[3] shows that a combination of over-sampling the minority class and under-sampling the majority class can achieve better classifier performance (in ROC space) than only under-sampling the majority class. Observing the behavior of the model with only oversampling, the accuracy was limited a mediocre value (Fig. 2), but when we integrated the under sampling of the majority classes, the accuracy shots up to a more significant percentage (Fig. 3)



```
----------------------------------
Accuracy of 6 vs Rest:  0.8021
Accuracy of 7 vs Rest:  0.7727
Accuracy of 8 vs Rest:  0.8449
Accuracy of 9 vs Rest:  0.8289
Accuracy of 10 vs Rest:  0.9893
----------------------------------
```

Fig. 2 Accuracies when only SMOTE was used.



```
----------------------------------
Accuracy of 6 vs Rest:  0.8846
Accuracy of 7 vs Rest:  0.9731
Accuracy of 8 vs Rest:  0.7923
Accuracy of 9 vs Rest:  0.8846
Accuracy of 10 vs Rest:  1.0
----------------------------------
```

Fig. 3 Accuracies when SMOTE and ENN was used.

We use SMOTE to generate the synthetic data instances and combined Edited Nearest Neighbours (ENN) to remove the majority class instance.

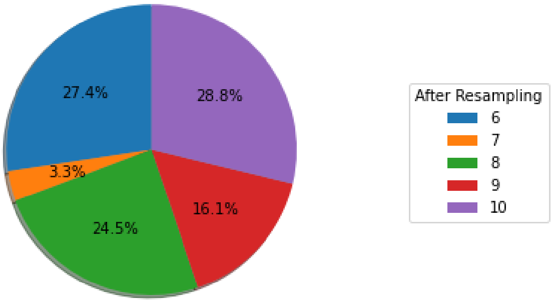


Fig. 4 Distribution of Gleason Score after resampling

Normalizing the Data

Before normalizing the data, we split the data into train and test set (7:3 ration). We find the minimum and maximum values for each feature from the train set, after which we transform the train and test set to fall under [0, 1] range. The main motivation to do normalization of the data was based on the ambiguity of distribution of the data.

C. Feature Selection

After some filtering while data preprocessing, we were able to reduce the dimensionality of gene dataset to 57470 features. This is still a huge number, and the curse of dimensionality makes it difficult to classify this dataset. Thus, we decided to use feature selection techniques to reduce the number of gene to a handful. Four different feature selection methods were explored, and the best ones were used for this purpose. Combination of wrapper feature selection and filter feature selection are used to improve the selection process. Chi-square and mutual information gain (filter feature selection) are applied to first select the couple hundred features. After that, forward and backward wrapper feature selection were explored to extract best subset of most important features. In the try to find the best wrapper feature selection method, we realized that backward feature selection, although took a very long time to run but gave us a very high accuracy. Below we briefly discuss the feature selection techniques we used:

Select K-Best

SelectKBest is a technique in which one can identify the K best features. Best can be defined as the ones that are highly correlated to the label. Correlation is a measure of how two variable changes together. Linear correlation scores are typically a value between –1 and 1 with 0 representing no relationship. For feature selection, we are often interested in a positive score. The larger the positive value, larger is the relationship, and, more likely, the feature should be selected for modelling. As such the linear correlation can be converted into a correlation statistic with only positive values. Select K-Best selects the top k most relevant features. It requires us to specify the score function. 2 score functions we used for our analysis were: chi-square and mutual gain info.

Filter feature selection

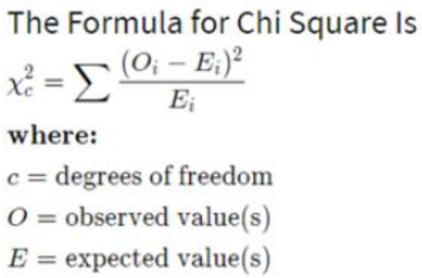
The features selection process is independent of classification algorithms, filter model relies on the general characteristics of the training data to select features with independence of any predictor. Filter methods are usually computationally less expensive than other feature selection paradigms. So, in those cases in which the number of features is very large, filter methods are indispensable to obtain a reduced set of features. In fact, this is the basis of most recent hybrid algorithms for feature selection. One of the problems that one might face while using the filter method is which filter gives the best relevance index. Since filter methods carry out the feature selection with no induction algorithm, the general characteristics of the training data are used to select the features like distances between classes or statistical dependencies. This helps the filter method to select a subset of the large set of features in a very short computational time.

1. Chi-square

A chi-square test is used in statistics to test the independence of two events. Given the data of two variables, we can get observed count O and expected count E. Chi-square measures how expected count E and observed count O deviates each other. In feature selection, we aim to select the features which are highly dependent on the response. When two features are independent, the observed count is closer to the expected count, thus we will have smaller Chi-Square value. So high Chi-Square value indicates that the hypothesis of independence is incorrect. In simple words, higher the Chi-Square value the feature is more dependent on the response, and it can be selected for model training.

Steps for Chi-2 Test:

a. Define Hypothesis.
b. Build a Contingency table.
c. Find the expected values.
d. Calculate the Chi-square statistic.
e. Accept or reject the Null Hypothesis.



The Formula for Chi Square Is

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

where:

c = degrees of freedom

O = observed value(s)

E = expected value(s)

Fig.5 Chi-Square

Limitation of Chi-Square Test has to do with the sensitivity to small frequencies in cells of table. Generally, when the expected value in a cell of a table is less than certain threshold, chi-square can lead to errors in conclusions.

2. Mutual Information Gain

This feature selection method is used primarily in the construction of decision trees from a training dataset, by evaluating the information gain for each variable, and selecting the variable that maximizes the information gain, which in turn minimizes the entropy and best splits the dataset into groups for effective classification.

Information Gain can also be used for feature selection, by evaluating the gain of each variable in the context of the target variable. In this slightly different usage, the calculation is referred to as mutual information between the two random variables. Mutual information calculates the statistical dependence between the two variables and is the name given to information gain when applied to variable selection.

Information Gain measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. A large information gain suggests a lower entropy group or groups of samples, and hence less surprise. In Information Theory, we like to describe the "surprise" of an event. Low probability events are more surprising therefore have a greater amount of information. Whereas probability distribution where the events are equally likely are equally likely are more surprising and have a larger entropy.

Wrapper Feature Selection

We were able to reduce 57470 features to 300 features using filter feature selection method. We then applied wrapper Feature selection to further reduce the features. Wrapper method of feature selection is based on **classification algorithm** that we are trying to fit on a given dataset. It follows a **greedy search approach** by evaluating all the possible combinations of features against the evaluation criterion. The evaluation criterion is simply the performance measure which depends on the type of problem. The fact that it used greedy search approach to find the best subset of feature, it becomes important to use filter feature selection prior to applying wrapper feature selection to improve the process of feature selection. Two techniques which we explored are:

1. Forwards Selection

In forward features selection model, we start with a null model and then start fitting the model with each individual feature one at a time and select the feature with the minimum p-value. Now fit a model with two features by trying combinations of the earlier selected feature with all other remaining features. Again, select the feature with the minimum p-value. Repeat this until we have a set of selected features with a p value of individual feature less than the significance level.

2. Backward Elimination

In backward elimination, we start with the full model (including all the independent variables) and then remove the insignificant features with the highest p-value (>significance level). This process repeats again and again until we have the final set of significant features.

As mentioned earlier both backward and forward feature selections depend on the classification algorithm and their accuracy. Every time a new feature is added (forward selection) or removed (backward selection), we calculate the resulting accuracy using 5-fold cross validation. The data is divided into 5 folds and then the learning is done on the 4 folds while the testing is done on the 1-fold which was left out. Idea is to use different combination of 4 folds every time we train the classifier. Thus, the classifier is now trained 4 times and the accuracy or other performance measure is then calculated.

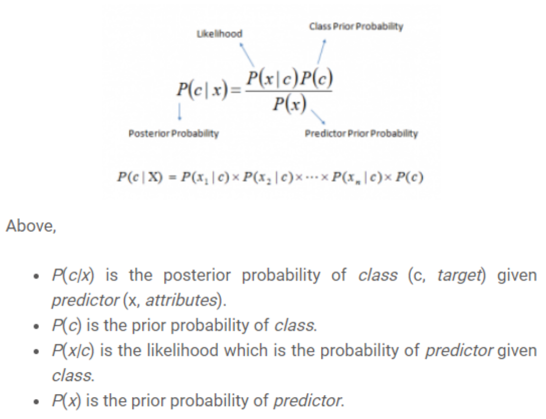
D. Classification

Since the problem falls under the multiclass classification problem, we used one-versus-rest technique and trained 5 different classifiers namely: 6 vs Rest, 7 vs Rest, 8 vs Rest, 9 vs Rest, 10 vs Rest. This process involves classifying one class against the remaining classes. Several standard classifiers like Naïve Bayes, LDA, SVM (rbf-kernel) and Random Forest were explored. The one which gave us the best results for all 5 classifiers was random forest combined with Mutual Gain information and Backward feature selection.

Naïve Bayes

It is an extremely fast classification algorithm relative to other classification methods. It works on the principle of Bayes theorem of probability to predict the class of unknown data sets. It is a classification technique based on Baye's Theorem with an assumption of independence among predictors. In simple terms, a Naïve bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. Even if these features depend on each other or upon the existence of the other features, all these properties independently contribute to the probability that the sample belongs to a certain class and that is why it is known as 'Naive'.

Naïve Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naïve Bayes is known to outperform even highly sophisticated classification methods.


Likelihood
Class Prior Probability
Posterior Probability
Predictor Prior Probability

$$P(c \mid x) = \frac{P(x \mid c) P(c)}{P(x)}$$

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

Above,

- *P*(*c*|*x*) is the posterior probability of *class* (c, *target*) given *predictor* (x, *attributes*).
- *P*(*c*) is the prior probability of *class*.
- *P*(*x*|*c*) is the likelihood which is the probability of *predictor* given *class*.
- *P*(*x*) is the prior probability of *predictor*.

Fig. 6 Naïve bayes formula

Steps for Naïve Bayes:

a. Convert the dataset into a frequency table.
b. Create likelihood table by finding the probabilities like Overcast probability.
c. Finally, use Naïve Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Pros: It is easy and fast to predict class of test dataset. It also performs well in multi class prediction. When assumption of independence holds, a Naïve Bayes classifier performs better compared to other models like logistic regression and you need less training data. It performs well in case of categorical input variables compared to numerical variables. For numerical variables, normal distribution is assumed.

Cons: If Categorical variable has a category (in test dataset), which was not observed in training dataset, then model will assign a 0 probability and will be unable to make a prediction. This is often known as “Zero Frequency.” To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation. On the other hand, naïve bayes is also known as a bad estimator, so the probability outputs from predict_proba are not to be taken too seriously. Another limitation of Naïve Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

Naïve Bayes is good for real time prediction, multi class prediction, text classification/ spam filtering / sentiment Analysis and Recommendation Systems with Collaborative Filtering.

Linear Discriminant Analysis

For more than two classes Linear Discriminant analysis is the preferred linear classification technique. LDA is a simple model in both preparation and application. There is some interesting statistics behind how the model is setup and how the prediction equation is derived. These statistical properties are estimated from your data and plug into the LDA equation to make predictions. These are the model values that you would save to file for your model. LDA makes some simplifying assumptions about your data:

1. That your data is Gaussian, that each variable is shaped like bell curve when plotted.
2. That each attribute has the same variance, that values of each variable vary around the mean by the same amount on average.

With these assumptions, the LDA model estimates the mean and variance from your data for each class. It is easy to think about this in the univariate case with two classes. The mean value of each input for each class can be estimated in the normal way dividing the sum of values by the total number of values.

LDA makes predictions by estimating the probability that a new set of inputs belong to each class. The cases that get the highest probability is the output class, and a prediction is made. The model uses Bayes Theorem to estimate the probability. LDA Classify both multi class and binary categorical problems. It also works very well on Normalized dataset (data which is in Gaussian Distribution), and which has no outliers.

LDA also assumes the fact that the input variables have the same variance.

Random Forest

Random forest is a model that uses combination of decision tress to make a decision. It counts the result of each tree as votes to decide the final class of the data instance. This system of voting helps it avoid the problem of overfitting. Several different decisions tree. In general, each vote from different decision tress is given equal weight. The way it works is as follows:

a. A data set is sampled with replacing the members from the training set. This process is called bootstrapping. This may result in duplicate examples from the training set. According to [4], usually one third of the training data is not present in the bag. This left-over data is often called out-of-bag data.

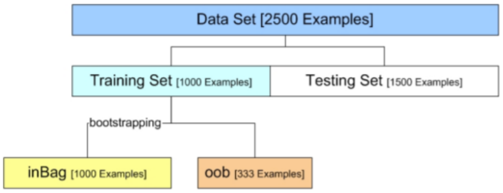


Fig 7. Samples with replacements (Livingston, F. (2005))

b. Then out of all the features, random numbers of features are chosen for each tree. A standard tree building algorithm is then used to form the tree.

c. We then each tree grows to its fullest.

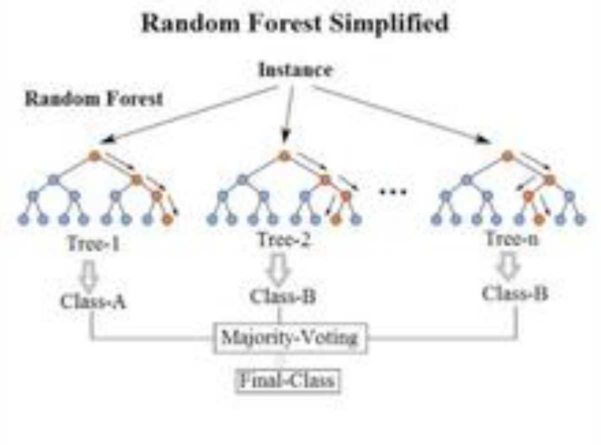


Fig. 8 Random Forest classifier

K Nearest Neighbor

K Nearest Neighbor is a supervised classification technique. This classifier works by calculating the distance from the given data points with the K nearest neighbor and assigns the labels to that data point equal to the majority amongst the K neighbors. It is one of the simplest classification techniques and we were able to get some amazing result using this method.

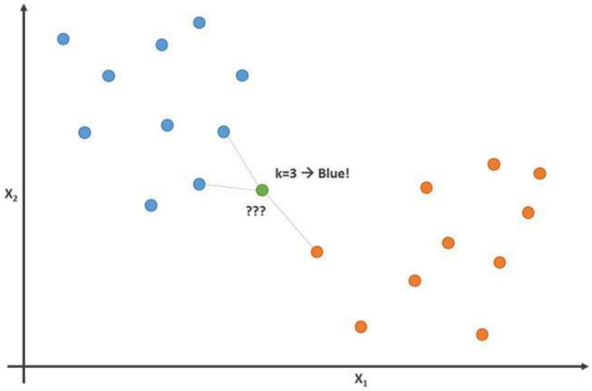


Fig. 9 KNN

Support Vector Machine

SVM is a kernel-based learning method. Kernel based learning methods have gained much attention in recent years due to their high performance. The idea is to implicitly transform the features to higher dimension where the data is linearly separatble[5]. A function φ: X → H mapping from an input space X to a feature space H is incorporated in a kernel via dot product:

$$k(\mathbf{v}_i, \mathbf{v}_j) = \langle \phi(\mathbf{v}_i), \phi(\mathbf{v}_j) \rangle.$$

Fig.10 Mapping function

We tried linear kernel and RBF kernel and received better results with RBF kernel. RBF kernels transform the feature in infinite dimension which results in higher accuracy. Results of the SVM-rbf can be seen in Table III.

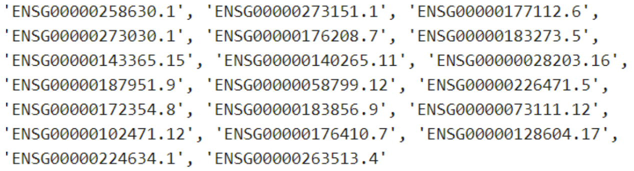
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

Fig. 11 RBF Kernel

III. RESULTS AND DISCUSSION

All the above 5 classifications techniques were used along with combination of feature selection techniques to train 5 one-versus-rest classifiers. Best results were obtained from KNN (mutual gain info + forward selection), SVM-rbf (mutual gain + forward selection) and random forest (mutual gain information + backward selection), but Random Forest outperformed all other models. Accuracies for each model are depicted in the table below:

A. Naïve Bayes



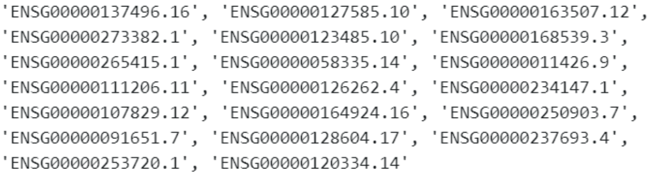
```
'ENSG00000258630.1', 'ENSG00000273151.1', 'ENSG00000177112.6',
'ENSG00000273030.1', 'ENSG00000176208.7', 'ENSG00000183273.5',
'ENSG00000143365.15', 'ENSG00000140265.11', 'ENSG00000028203.16',
'ENSG00000187951.9', 'ENSG00000058799.12', 'ENSG00000226471.5',
'ENSG00000172354.8', 'ENSG00000183856.9', 'ENSG00000073111.12',
'ENSG00000102471.12', 'ENSG00000176410.7', 'ENSG00000128604.17',
'ENSG00000224634.1', 'ENSG00000263513.4'
```

Fig 12. Selected Genes (Mutual Gain + Forward Selection)

6 vs rest	76.54%
7 vs rest	60.77%
8 vs rest	57.69%
9 vs rest	44.62%
10 vs rest	98.08%

Table I. Naïve bayes accuracy

B. Linear Discriminant Analysis



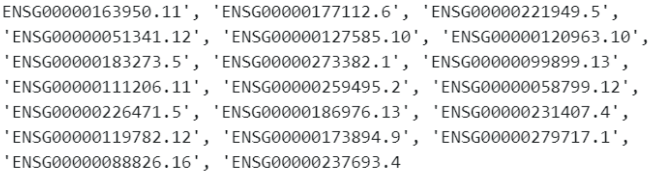
```
'ENSG00000137496.16', 'ENSG00000127585.10', 'ENSG00000163507.12',
'ENSG00000273382.1', 'ENSG00000123485.10', 'ENSG00000168539.3',
'ENSG00000265415.1', 'ENSG00000058335.14', 'ENSG00000011426.9',
'ENSG00000111206.11', 'ENSG00000126262.4', 'ENSG00000234147.1',
'ENSG00000107829.12', 'ENSG00000164924.16', 'ENSG00000250903.7',
'ENSG00000091651.7', 'ENSG00000128604.17', 'ENSG00000237693.4',
'ENSG00000253720.1', 'ENSG00000120334.14'
```

Fig 13. Selected Genes (Chi2 + Forward Selection)

6 vs rest	90.0%
7 vs rest	96.15%
8 vs rest	72.31%
9 vs rest	89.62%
10 vs rest	98.46%

Table II. LDA accuracy

C. Support Vector Machine – RBF



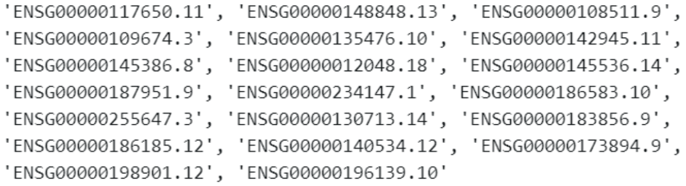
```
ENSG00000163950.11', 'ENSG00000177112.6', 'ENSG00000221949.5',
'ENSG00000051341.12', 'ENSG00000127585.10', 'ENSG00000120963.10',
'ENSG00000183273.5', 'ENSG00000273382.1', 'ENSG00000099899.13',
'ENSG00000111206.11', 'ENSG00000259495.2', 'ENSG00000058799.12',
'ENSG00000226471.5', 'ENSG00000186976.13', 'ENSG00000231407.4',
'ENSG00000119782.12', 'ENSG00000173894.9', 'ENSG00000279717.1',
'ENSG00000088826.16', 'ENSG00000237693.4
```

Fig 14. Selected Genes (Mutual + Forward Selection)

6 vs rest	93.85%
7 vs rest	97.31%
8 vs rest	89.62%
9 vs rest	95.38%
10 vs rest	100.0%

Table III. SVM accuracy

D. K Nearest Neighbours (3 Neighbours)



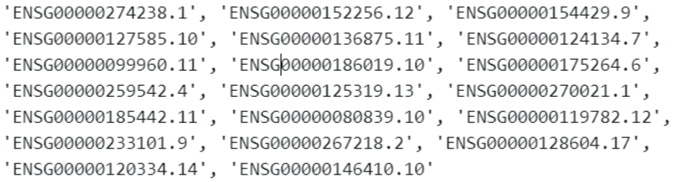
```
'ENSG00000117650.11', 'ENSG00000148848.13', 'ENSG00000108511.9',
'ENSG00000109674.3', 'ENSG00000135476.10', 'ENSG00000142945.11',
'ENSG00000145386.8', 'ENSG00000012048.18', 'ENSG00000145536.14',
'ENSG00000187951.9', 'ENSG00000234147.1', 'ENSG00000186583.10',
'ENSG00000255647.3', 'ENSG00000130713.14', 'ENSG00000183856.9',
'ENSG00000186185.12', 'ENSG00000140534.12', 'ENSG00000173894.9',
'ENSG00000198901.12', 'ENSG00000196139.10'
```

Fig 15. Selected Genes (Mutual Info + Forward Selection)

6 vs rest	96.54%
7 vs rest	96.92%
8 vs rest	93.46%
9 vs rest	93.46%
10 vs rest	98.85%

Table IV. KNN(3-neighbours) accuracy

E. Random Forest



```
'ENSG00000274238.1', 'ENSG00000152256.12', 'ENSG00000154429.9',
'ENSG00000127585.10', 'ENSG00000136875.11', 'ENSG00000124134.7',
'ENSG00000099960.11', 'ENSG00000186019.10', 'ENSG00000175264.6',
'ENSG00000259542.4', 'ENSG00000125319.13', 'ENSG00000270021.1',
'ENSG00000185442.11', 'ENSG00000080839.10', 'ENSG00000119782.12',
'ENSG00000233101.9', 'ENSG00000267218.2', 'ENSG00000128604.17',
'ENSG00000120334.14', 'ENSG00000146410.10'
```

Fig 16. Selected Genes (Mutual Info+ Backward Selection)

6 vs rest	98.08%
7 vs rest	96.92%
8 vs rest	95.77%
9 vs rest	95.77%
10 vs rest	100.0%

Table V. Random Forest accuracy

Table V. shows that for each 5 classifiers the model generated using by Random Forest gave a very good accuracy. SVM and KNN were also able to produce some great results. We saw a constant improvement in accuracies when experimenting with classification techniques like, Naïve bayes, LDA, SVM, K nearest Neighbor and Random Forest which can be noticed in the accuracy tables given above. We also noticed that using the subset of features chosen by backward feature selection we were able to drastically improve the model performance. Using the dataset that initially had more than 60000 genes, our method was able to identify 20 most important genes that contribute to defining the Gleason score of a patient. The 20 genes can now be used as biomarkers in treating Prostate cancer.

IV. CONCLUSION

Since the advancement of Machine Learning since several decades, it has been used in cancer research. This step of discovering the gene biomarkers for prostate cancer is a significant step in determining the proper treatment for patients. We hope that using our model, doctors can now easily estimate the Gleason score of a patient. This score can then be used to predict the behavior of a prostate cancer based on which proper treatment can be provided to a patient.

V. FUTURE WORK

Future work includes doing research and biological Analysis of these 20 genes to find their biological relevance. One could also try using some other feature selection techniques like mRMR and try training a model based on the selected genes. We could not find any better implementation of mRMR and thus only tried forward and backward selection methods.

VI. AUTHOR CONTRIBUTION

Aman - Data cleaning and preprocessing, Experimenting with Naïve Bayes, LDA and KNN
Aditya – Experimented with Dimensionality Reduction (Although he noticed poor accuracy and choose not to include it), SVM-rbf, Random Forest.
Most of the work was doing together during pair programming. All authors have equal contribution in brainstorming and writing the report.

VII. REFERENCES

[1] Key Statistics for Prostate Cancer: Prostate Cancer Facts, https://www.cancer.org/cancer/prostate-cancer/about/key-statistics.html.

[2] Gleason Score, https://www.prostateconditions.org/about-prostate-conditions/prostate-cancer/newly-diagnosed/gleason-score

[3] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357. doi: 10.1613/jair.953

[4] Livingston, F. (2005). Implementation of Breiman's random forest machine learning algorithm. ECE591Q Machine Learning Journal Paper, 1-13.

[5] Chudzian P. (2011) Radial Basis Function Kernel Optimization for Pattern Classification. In: Burduk R., Kurzyński M., Woźniak M., Żołnierek A. (eds) Computer Recognition Systems 4. Advances in Intelligent and Soft Computing, vol 95. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-20320-6_11