Personalized Cancer Diagnosis using Machine Learning

Karun Papreja* Sachin * Yashu Chaudhary*Brijal*

*Centre for Technology Alternatives for Rural Areas (CTARA), IIT Bombay
*Environmental Science & Engineering Department, IIT Bombay

Abstract

Cancer is the second leading cause of death globally and accounted for 8.8 million deaths in 2015. It has been characterized as a heterogeneous disease consisting of many different subtypes. The early diagnosis and prognosis of a cancer type have become a necessity in cancer research, as it can facilitate the subsequent clinical management of patients. For better clinical decisions, it is important to accurately distinguish between benign and malignant tumors. Conventionally, statistical methods have been used for classification of high risk and low risk cancer, despite the complex interactions of high-dimensional medical data. To overcome the drawbacks of conventional statistical methods, machine learning has emerged as a promising technique for handling high-dimensional data, with increasing application in clinical decision support. This paper highlights new research directions and discusses main challenges related to machine learning approaches in cancer detection. Basically we are trying to reduce the time consumption in classifying the type of tumor using Multiple machine learning models such as Naïve Bayes, Linear SVM, KNN, logistic Regression, Random Forest.

Libraries: We are using multi class classification problem. We are using libraries such as Pandas, matplotlib, nltk, scikitlearn, NumPy etc. for this project. We are getting the real world data of cancer patients from a US website. Based on that data, we are going to get the probability for the cancer diagnosis. The early diagnosis of cancer can improve the prognosis and chance of survival significantly, as it can promote timely clinical treatment to patients. So,by using the data, we will try to get the probability of cancer diagnosis.

Keywords: - Cancer Detection, Machine Learning, Classification, Prediction

INTRODUCTION: - Our body is made up of trillions of cells, which are constantly dying and regenerating. Normally, a cell divides and make a perfect copy of itself using a genetic blueprint called DNA. Once in a while, the DNA blueprint gets damaged sometimes so that cell doesn't listen to body signals and keeps on dividing forming a tumor. When a patient seems to have cancer, we take a tumor sample from the patient and we go through genetic sequencing of DNA. Once sequenced, a tumor can have thousands of genetic mutations. Here briefly, a

‘mutation’ is small change in gene which causes cancer. One more important thing is that for every gene, there is a variation associated with it. Actually, molecular pathologist selects a list of genetic variations of interest that he/she want to analyze. The molecular pathologist searches for evidence in the medical literature that somehow are relevant to the genetic variations of interest. Finally, the molecular pathologist spends a huge amount of time analyzing the evidence related to each of the variations to classify them. However, selection of list of genetic variations and searching for evidence in medical literature can be done easily and with less time. But analyzing the evidence related to each type of variation to classify them is very time consuming. Our goal is to replace classification of cancer after analyzing the evidence related to every variation with a machine learning model.

OBJECTIVE: -

To predict the probability of each data point belonging to each of the 9 classes.

MOTIVATION: -

This kind of project can help in reduction in time and effort consumption in analyzing the evidence related to each type of variation, classifying them and predicting the probability of each data point.

PROBLEM STATEMENT: -

Classify genetic variations based on evidence from the text-based clinical literature or research papers using multi class classification among the 9 classes and classify which class it belongs to.

BUSINESS CONSTRAINTS OF THIS PROBLEM:

1. Interpretability of the algorithm is must because a cancer specialist should understand why the model is given particular class so that he can explain to the patient.
2. No low-latency requirement which means patient can wait for the results. As there is no low-latency requirement, we can apply complex machine learning models.
3. Errors are very costly.
4. Probability of belonging to class is needed rather than it belonging to particular class.

Methodology: -

We will go through the basic literature out there for implementation of multi-class classification and understand different techniques that one can employ. There can be multiple Performance metric like Area under curve, Precision, Recall, log loss, F1score, confusion metric. Performance metric are also known as KPI (Key Performance Indicator). It is very important to choose the right metric. So, we have chosen multi-class log loss because of our business constraints, our aim was to predict probabilities and for the same, the best model is multi-class Log loss and also for the reason because we want to penalize the model for error since errors are very costly in this scenario. For Better visual interpretation result, we have used confusion matrix.

Steps used are:

Step 1: Literature review: Collect papers using Scopus and google scholar.

Step 2: Based on the Literature review, we started defining our project research topic and objectives.

Step 3: Acquiring the databases

Step 4: Analyze the databases and reading data & preprocessing of data

Step 5: Splitting the Data into train, cross validation & test data

Step 6: Chose different metrics for trying out with different machine learning models which are more interpretable such as Naïve Bayes, Random Forest, Linear SVM. Logistic Regression, KNN.

Working: -

ID, gene, variation, text is my xi and class is my yi.

We even checked precision and recall matrix in which diagonal elements (which are precision and recall of all classes) seemed to be very low because of a random model. Precision and recall matrices are attached below. From the above distributions it is also very clear that 1,2,4,7 classes are majority classes, class 8,9,3 are least dominant. Data is imbalanced. Train, Test, CV data should have similar distribution.

Suppose we have a test Dataset.

$D_{test} = \{xi, yi\}_{i=1}^{n}$ where yi €{1,2,….9}

Average Multiclass Log Loss =

$$\frac{-1}{n}\sum_{i=1}^{n}\sum_{j=1}^{C=9} y'ij * logpij$$

Where i corresponds to points and j responds to classes.

y’ij= 1 if I £ class j

Or else 0

pij is the probability that the ith point belongs to class j for a input xi with a particular ML model.

Pre-Processing of data: - After reading the data, we preprocessed the data like removing stop-words, converting to lower case and removing punctuations etc. Preprocessing is a very imp. stage.

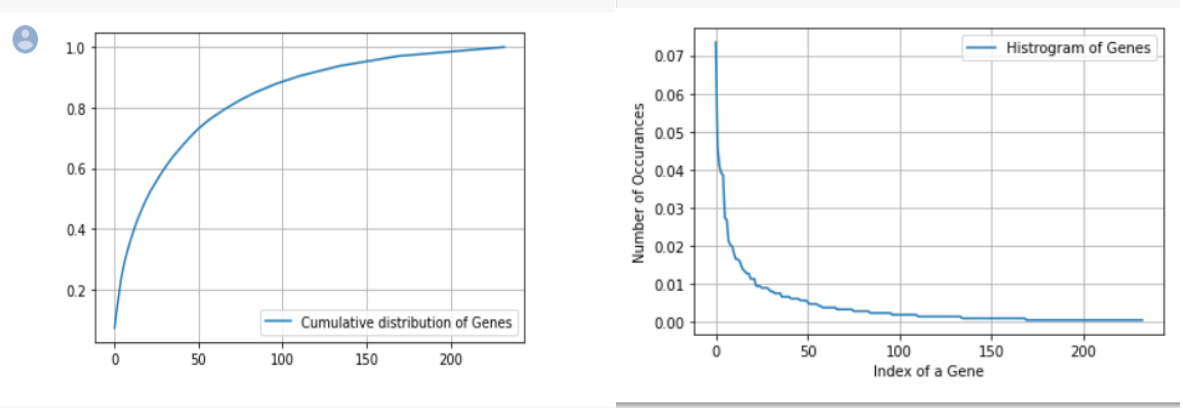
Splitting the data: As the data is not temporal in nature which means it is not changing with time we can split the data randomly for training, cross validation and testing in (64%, 16 %, 20%) respectively.

As we know that log-loss is ranging from 0 to infinite, so we first define a random model so that we can define a first random model, then we can consider that our ML model is good. After giving data to our random model, it gives a roughly log-loss of 2.5.

UNIVARIATE ANALYSIS: -

We take each feature and check whether it is useful for predicting in class label by various ways so that we can use that feature. If it is not useful, we can simply remove that feature

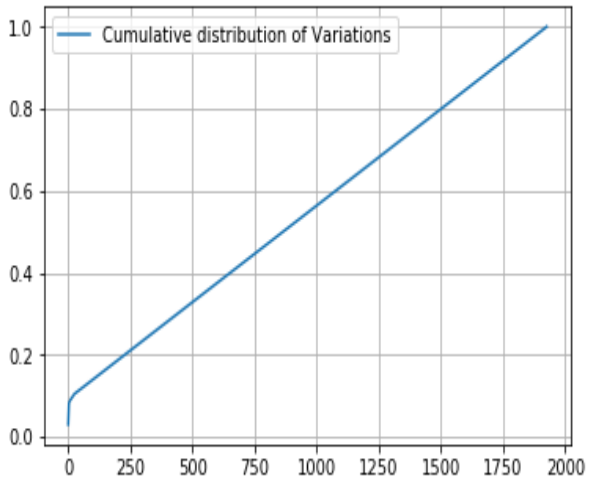
1. Gene Feature: -As we know that gene is a categorical feature. From that we observe that there are 235 types of unique genes out of which top 50 most frequent genes nearly contribute to 75 percent of data. If I look at histogram of data, it seems as a skewed distribution. The topmost gene occurs 8% of the time.



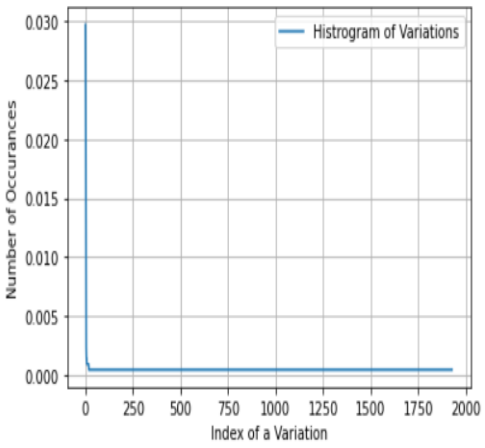
Now we feature the gene into vector by one hot encoding and response coding. Then we build one simple Logistic Regression model with Calibrated Classifier and applied gene feature and class labels to it. We find that Train, CV, Test log-loss values are roughly same and also find out that log loss value is less than 2.5(Random Classifier value). Hence we can say that Gene is an important feature for our classification. We can also conclude that gene is stable feature because CV and Test errors are roughly equal to Train Errors.

2. Variation Feature: - Here variation is also a categorical feature and we observed that 1927 unique variations out of 2124 present in training data which means most of variations occurred once or twice.

CDF of variations looks as follows:



Histogram of variations looks as follows:



Cumulative Distribution is straight line which means most variations occur once or twice in training data. We feature the Variation into vector by one hot encoding and response coding. As we did earlier for the gene feature, we build a simple LR model and apply data to it and find that log loss values of Train, CV, Test found to be less than Random Model. But the difference between Train log loss and CV, Test log loss is significantly more than gene feature which means variation feature is unstable. But as the log loss is less than the Random Model we still use the variation feature but be careful since it is not stable.

3. Text Feature: - In text data there are total 53,000 unique words which are present in training data. We also observe that most words occur very few times which is common in text data. We convert the text data into vector by BOW (one hot encoding) and Response Coding.

As we did in previous cases we apply it to simple model LR and log loss values of Train, CV, Test are found to be less than Random Model. From the distributions of CV, Test data it is found out that test feature is a stable feature.

Now combine all the features by two ways

1. One hot encoding: It is found out that by one hot encoding the dimensionality is 55,517 which is because of text data.

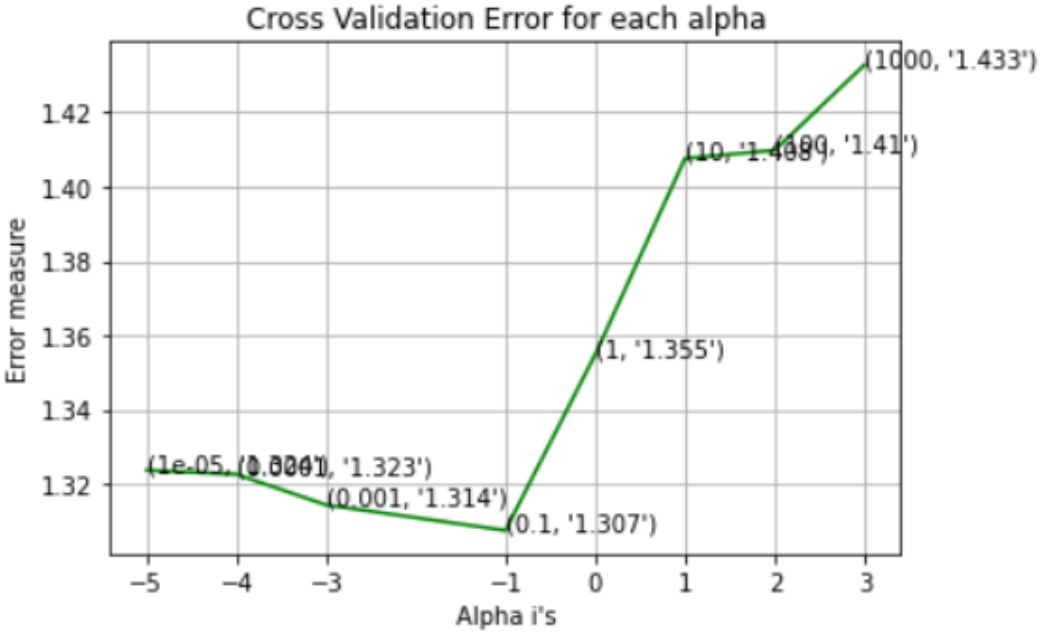
2. Response Coding: It is found out that by Response Coding the dimensionality is 27 (each feature corresponds to 9 dimensions).

BASELINE MODEL

ML model 1:

NAIVE BAYES:

We know that for text data NB model is a baseline model. Now we apply the training data to the model and used the CV data for finding best hyper-parameter(alpha)



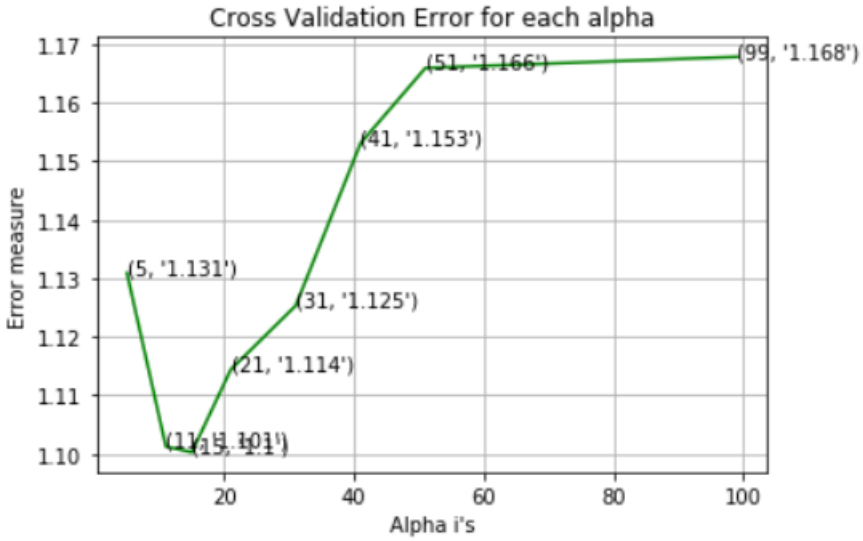
With the best alpha we fit the model. The test data is then applied to the model and we found

out that the log-loss value is 1.30 which is quite less than Random Model. Here we also find out that the total number of mis-classified cases is 41.54 percent. We also checked the probabilities of each class for each data and interpreted each point. This is to check why it is predicting particular class randomly. We conclude that for mis-classified points, the probability that the point belongs to a predicted class is very low.

APPROACH 2:

K Nearest Neighbors:

As we know that the k-NN model is not interpretable(which is our business constraint) but we still use this model just to find out the log loss values. Since k-NN suffers from the curse of dimensionality, we use response coding instead of one-hot encoding. After applying the data to the model we obtain the best hyper-parameter(k)



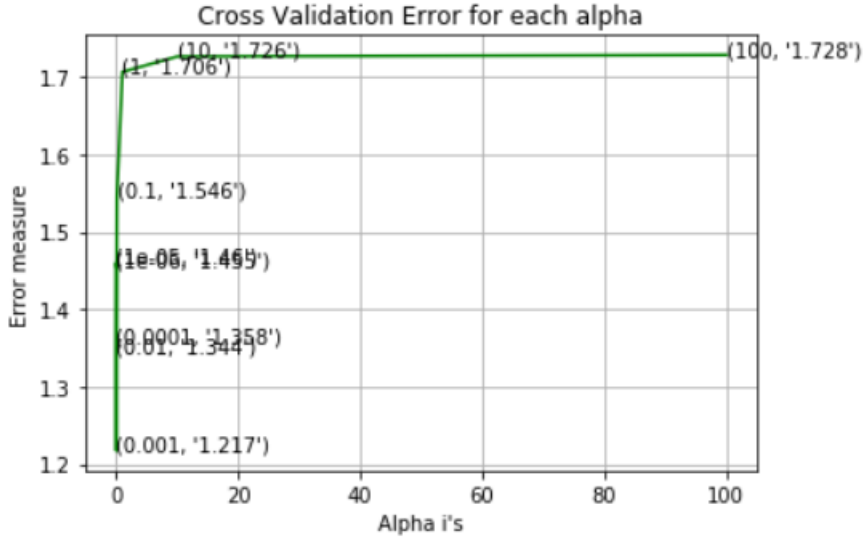
With the best k we fit the model and test data is applied to the model. The the log-loss value is 1.10 which is less than NB model. But number of mis-classified points are 39.66 percent (almost equal to NB model).

ML model 3:

3.LOGISTIC REGRESSION:

As we have already seen, the LR model worked very well with univariate analysis. So we did some thorough analysis on LR by taking both imbalanced data and balanced data. With Class Balancing: We also know that LR works well with high dimension data and it is also

interpretable. So we did oversampling of lower class points and applied the training data to the model and used the CV data for finding best hyper-parameter (lambda)

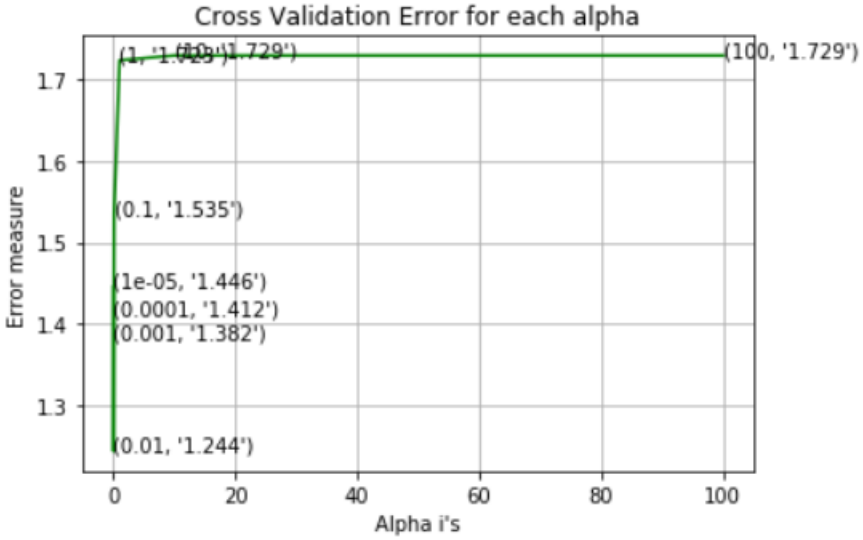


With the best lambda we fitted the model and test data is applied to the model. The log-loss value is 1.21. But number of mis-classified points are 38.34 percent As LR is interpretable and mis-classified points are less than other models(k-NN and NB)it is better than k-NN and NB. Without class balancing log loss and mis-classified points are increased. Therefore, we use class balancing.

ML Model 4:

4. SVM:

We use Linear SVM(with class balancing) because it is interpretable and works very well with high dimension data. RBF Kernel SVM is not interpretable so we cannot use it. Now we apply the training data to the model and use the CV data for finding best hyper-parameter (C)



With the best C we fit the model and test data is applied to the model. Now, the log-loss value is 1.24 which is quite less than Random Model. Here, the total number of mis-classified cases is 39.47 percent(more than LR). Since we used class balancing, we got good performance for minor classes.

ML model 5:

RANDOM FOREST:

5.1) One-hot encoding: Normally Decision Tree works well with low-dimension data. It is also interpretable. By changing the number of base learners and max depth in Random Forest Classifier, we get best base learners=2000 and max depth=10.Then we fit the model with best hyper-parameters and test data is applied to it. The resultant log loss value is 1.174(and total number of mis-classified points is 38.53 percent

5.2) Response Coding: By changing the number of base learners and max depth in Random Forest Classifier we find that best base learners=100 and max depth=5. We then fit the model with best hyper-parameters and found that train log loss is 51.87 percent, and CV log loss is 1.417 which says that model is overfitted even with best hyper-parameters. That is why we don't use RF+Response Coding.

Reference:

1. We are using the data from www.kaggle.com ,
Research paper: Applications of Machine Learning in Cancer Prediction and Prognosis.

2. Noor, M. M., & Narwal, V. (2017). Machine learning approaches in cancer detection and diagnosis: mini review. *IJ Mutil Re App St*, *1*(1), 1-8.

3. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V., & Fotiadis, D. I. (2015). Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal*, *13*, 8-17.