International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 04 | Apr 2020 www.irjet.net p-ISSN: 2395-0072

Personalized Medicine: Redefining Cancer Treatment

Samruddhi Mhatre [1*], Sejal Rai [2*], Shraddha Waghukar[3*], Prof. Sonali Dhamele[4]

[1]*Final Year Student, Department of Computer Engineering, Terna Engineering College, Nerul, India.*
[2]*Final Year Student, Department of Computer Engineering, Terna Engineering College, Nerul, India.*
[3]*Final Year Student, Department of Computer Engineering, Terna Engineering College, Nerul, India.*
[4]*Professor, Department of Computer Engineering, Terna Engineering College, Nerul, India.*
**All Authors have Contributed Equally*

Abstract - *Establishing a diagnosis through exome sequencing can provide potential benefits to patients, insurance companies, and the healthcare system. Genetic data is being produced in vast amounts which needs careful organization. Hence, diagnostic sequencing is being employed increasingly. We have discussed various machine learning methods which accurately assess the clinical validity of gene-disease relationships to interpret new research findings in a clinical context which in turn increases the diagnostic rate. The solution to deal with huge data effectively is by using Machine learning. Medical science yields huge amount of data on daily basis from research and development (R&D), physicians and clinics, patients, care givers etc. Diagnosis via machine learning works when the condition can be reduced to a classification task on physiological data, in areas where we currently rely on the clinician to be able to visually identify patterns that indicate the presence or type of the condition. A lot has been said during the past several years about how precision medicine and, more concretely, how genetic testing is going to disrupt the way diseases like cancer are treated. But this is only partially happening due to the huge amount of manual work still required. This project aims at building a machine learning model to take personalized medicine to its full potential.*

***Key Words*: Genetic Mutation, Cancer Treatment, Machine Learning, Text Classification, Gene, Variation.**

1. INTRODUCTION

A gene is a sequence of nucleotides in DNA or RNA that encodes the synthesis of a gene product, either RNA or protein. Nucleotide is made up of 3-component: Phosphate, Sugar and Nitrogen Base. The five bases are adenine, guanine, cytosine, thymine, and uracil, which have the symbols A, G, C, T, and U, respectively. Difference in DNA among individuals is known as Genetic Variations. There are multiple sources of genetic variation, including mutation and genetic recombination. A mutation is the alteration of the nucleotide sequence of the genome of an organism. Once sequenced, a cancer tumour can have thousands of genetic mutation. The interpretation of genetic mutations is a very time-consuming task since it is being done manually.

Accurate prediction of survival in patients with cancer remains a challenge due to the ever-increasing heterogeneity and complexity of cancer, treatment options and patient populations. In current practice, clinicians use data collected at the bedside in consultations, medical records or purpose-built cancer registries to aid prognostication and decision-making. If achieved, reliable predictions could assist personalized care and treatment, and improve institutional performance in cancer management.

The project aims at developing a Machine Learning algorithm that uses expert-annotated knowledge base where world-class researchers and oncologists have manually annotated thousands of mutations as a baseline, to automatically classify genetic variations.

Gene sequencing has rapidly moved from the research domain into the clinical setting. In the past couple of years, a lot of research efforts are concentrated on genetically understanding the disease and selecting the treatment that is most suited to the patients. Genetic testing is one of the innovative precision medicine technique and the way diseases like Cancer are treated. The major hurdle in using gene classification is a huge amount of manual work is still required. Memorial Sloan Kettering Cancer Center (MSKCC) launched a competition, accepted by the NIPS 2017 Competition Track, as help was required to take personalized medicine to its full potential. MSKCC is a cancer treatment and research institute in New York. It is the largest and oldest private cancer center in the world.

A Cancer tumor can have thousands of genetic mutations. The aim is to distinguish the mutations which contribute to tumor growth from neutral mutations.

2. LITERATURE REVIEW

Numerous researches have been carried out in this research area.

1. Machine learning applications in cancer prognosis and prediction [1] authored by Konstantina Kourou, Themis P. Exarchos, Konstantinos P. Exarchos, Michalis V. Karamouzis, Dimitrios I. Fotiadis.

This paper discusses the categorization of cancer 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. The importance

of classifying cancer patients into high or low risk groups has led many research teams, from the biomedical and the bioinformatics field, to study the application of machine learning (ML) methods. Therefore, these techniques have been utilized as an aim to model the progression and treatment of cancerous conditions. In addition, the ability of ML tools to detect key features from complex datasets reveals their importance. A variety of these techniques, including Artificial Neural Networks (ANNs), Bayesian Networks (BNs), Support Vector Machines (SVMs) and Decision Trees (DTs) have been widely applied in cancer research for the development of predictive models, resulting in effective and accurate decision making.

2. Applications of Machine Learning in Cancer Prediction and Prognosis[2] by Joseph A. Cruz, David S. Wishart.

In this paper a number of trends are noted in cancer prediction, including a growing dependence on protein biomarkers and microarray data, a strong bias towards applications in prostate and breast cancer, and a heavy reliance on "older" technologies such artificial neural networks (ANNs) instead of more recently developed or more easily interpretable machine learning methods. A number of published studies also appear to lack an appropriate level of validation or testing. Among the better designed and validated studies it is clear that machine learning methods can be used to substantially (15-25%) improve the accuracy of predicting cancer susceptibility, recurrence and mortality. At a more fundamental level, it is also evident that machine learning is also helping to improve our basic understanding of cancer development and progression.

3. Technologies for deriving primary tumor cells for use in personalized cancer therapy.[3] authored by Mitra A1, Mishra L, Li S.

This review focuses on our current understanding and the pros and cons of different methods for primary tumor cell culture. Furthermore, various culture matrices such as biomimetic scaffolds and chemically defined media supplemented with essential nutrients, have been prepared for different tissues. These well-characterized primary tumor cells redefine cancer therapies with high translational relevance.

4. Breast Cancer Prediction and Detection using Data Mining Classification Algorithms a Comparative Study.[4]. This paper, aims to predict and detect breast cancer early with non-invasive and painless methods that use data mining algorithms.

3. METHODOLOGY

In this project, we have developed a Machine Learning algorithm which makes use of expert annotated knowledge base as a baseline, which automatically classifies genetic variations. Our approach to the machine learning aspect of this project is to try few classification techniques and feature combinations as possible – such that each method is likely to bring a new perspective which captures features unique to each class.

First step is to visualize the data and extract as much information as possible. Then we have to examine the relationships between genes and classes and since it is known that one gene could fall into many different classes, suggesting that mutations within the same gene could produce widely different effects. It is known that the majority of mutations in each case are point mutations, in which one amino acid is mutated to another. It is to be noted that the genes included in the training and testing datasets were almost entirely different, and since the gene/mutation datasets contains limited information, we acquired most of our information from the text data.

The project aims at developing a Machine Learning algorithm that uses expert-annotated knowledge base where world-class researchers and oncologists have manually annotated thousands of mutations as a baseline, to automatically classify genetic variations. The Various Machine Learning Model used is as follows:

i. Naïve Bayes
ii. K-Nearest Neighbors
iii. Logistic Regression
iv. Linear Support Vector Machine
v. Random Forest Classifier
vi. Stacking Model

In our research, the task was to classify genetic mutations to enable personalized medicine for cancer treatment. The data comes from publicly available source provided by the Kaggle competition named "Personalized Medicine: Redefining Cancer Treatment". Due to the text-based nature of given dataset , a conversion was needed for any classification algorithms to be adopted. We used TF-IDF, one hot encoding and label encoding techniques for feature extraction and conversion; we used the above classification models to compare the performance. The evaluation of two classification methods was done based on multi class log loss, which is specified by the Kaggle website.

3.1 DATASET

The collected our dataset from the Kaggle competition and Memorial Sloan Kettering Cancer Center (MSKCC). The dataset has three parameters: genes, variations and clinical text. In the training set, 9 classes of mutations are given. Our goal was to use these three provided variables to predict mutation classes. The training set has 3321 data entries which are about four times as much as the test set. A similar distribution can also be seen with the types of variations. However, there were more types of genes in the test set compare with the number of types in the training set.

Both training and testing data sets are provided via two different files. The variants file provides information about the genetic mutations, whereas the test data file provides the clinical evidence in text format that pathologists use to classify the genetic mutations. Both the data files are linked via the ID field.

The genetic mutation with ID = 15 in the file training_variants can be classified using the clinical evidence in text format with ID = 15 in the file training_text.

File Descriptions:

training_variants - a comma separated file containing the description of the genetic mutations used for training. Fields are an ID (the id of the row used to link the mutation to the clinical evidence), Gene (the gene where this genetic mutation is located), Variation (the amino acid change for this mutations), Class (1-9 the class this genetic mutation has been classified on)

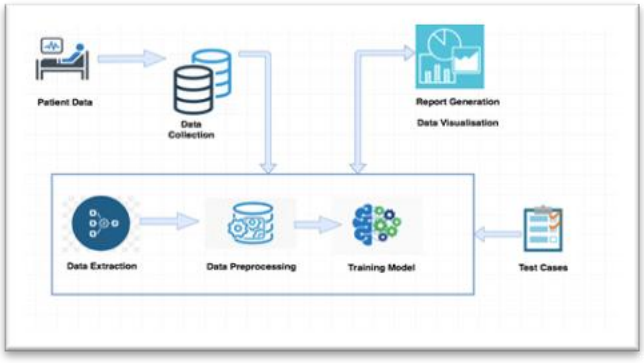
training_text - a double pipe (||) delimited file that contains the clinical evidence (text) used to classify genetic mutations. Fields are an ID (the id of the row used to link the clinical evidence to the genetic mutation), Text (the clinical evidence used to classify the genetic mutation)

test_variants - a comma separated file containing the description of the genetic mutations used for training. Fields are an ID (the id of the row used to link the mutation to the clinical evidence), Gene (the gene where this genetic mutation is located), Variation (the amino acid change for this mutations.

test_text - a double pipe (||) delimited file that contains the clinical evidence (text) used to classify genetic mutations. Fields are an ID(the id of the row used to link the clinical evidence to the genetic mutation), Text (the clinical evidence used to classify the genetic mutation)

3.2 APPROACH

Our approach is illustrated below.



The above sequence diagram describes the steps used to implement the project. Initially, raw data is gathered and analyzed to gain useful insights by performing Exploratory Data Analysis on the raw data. This data is further processed in order to reduce the data size and keeping only the useful information which help the training model in making accurate predictions. Once a training model is used it is tested against the testing data set to verify the accuracy of the model.

Gene has a particular sequence and if any variation is observed, there are chances of developing a benign or cancerous tumor. To solve the problem, this process of predicting the class has to be automated using machine learning algorithms.

3.2.1. Data Analysis

We have two training files i.e., training_text, training_variants.
1. Training_text has an ID, TEXT columns
2. Training_variants has an ID, GENE, VARIATION, CLASS columns

The training_text has 3321 rows and training_variants has 3321 rows. The training_variants dataset has 9 unique classes that would be predicted based on Gene, Variation, and Text columns. The output is discrete so it's a Multi-Class Classification problem.

While building a model for a medical problem errors need to be minimized and to decrease the error component and answer the prediction in probability terms as required in the problem description more evidence will be needed to reduce ambiguity and get the most probable class. While predicting the Class for the patient, we can also attach the reason to the Class result explaining the reasons for our prediction. Hence, interpretability will be an evident component of this prediction.

3.2.2. Data Preprocessing

To achieve the models, the Text had to be pre-processed using Natural Language Toolkit (NLTK) so that it can be fed to the Machine Learning algorithm. The Text component actually contains the research papers relevant to every class that was predicated in that case manually. The Text, therefore, has a lot of numbers and stop-words. Since it is a research paper there also might be inappropriate spaces that need to be dealt with. These unnecessary portions were removed using the NLTK library which is a platform used for Python programs that work with human language data for applying in Statistical Natural Language Processing (NLP). It consists of Text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

We processed the Text by removing numbers, inappropriate spaces and converting it into the lower case to avoid errors using Regex.

3.2.3. Merging the Data Files

Once, the Text is processed we merged the train_variants and train_text data to achieve a result set comprising ID, GENE, VARIATION, CLASS, TEXT columns.

3.2.4. Cleaning the Data and Handling the Missing Values

The result set was analyzed to determine any missing values so that they could be handled and won't cause disruptions in the final predictions. There were only five missing values which could be handled in two ways:

i. Removing the rows
ii. Using Imputation (Replacing the null values)

It was decided to handle the missing data using Imputation by replacing the null values with the concatenation of GENE and VARIATION column. The data were cross-checked for null values after imputation.

3.2.5. Creating Train, Test and Cross-Validation Set

Before the data was split, all the spaces in the Gene and Variation column were replaced by _. The data was split into training, testing, and validation because when hyperparameter tuning is done, we try to improve accuracy with respect to Test set which can sometimes lead to bad models. The result data set was, therefore, first divided into an 80% and 20% split of Train and Test and then the 80% of Train was further divided into 80% and 20% for the final Train and Validation set.

3.2.6. Data Distribution in Train, Test and Cross-Validation Set

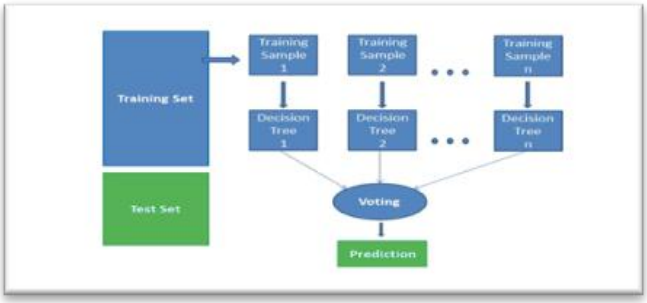
It is important that the distribution of data is similar in all three sets for good predictions. It was verified that the distribution of all these sets was almost similar.

3.2.7. Implementation Strategy

As log-loss is being considered as the main evaluation parameter a random model was generated to compare the log-loss against. Any successful model would return a lower log-loss than this random Model.

To evaluate the log-loss, of the cross-validation data for the random Model, an array of zeros was generated having the same length as the cross-validation data. The random probability was generated for all nine Class values in every row and stored in this array with respect to length.

Finally, an array of predicted Class values was generated, and the Confusion, Precision, and Recall Matrix were generated.



After calculating log-loss for all the considered models, we found that the log-loss for Random Forest Model was the lowest.

Therefore, to build the system we used the Random Forest Model.

3.2.8. Data Evaluation Concepts

Independent Columns: Gene, Variation, Text
Dependent Columns: Class [values from 0-9]

Next step is to check, how the gene is impactful on the class column. And if it is, categorical information of columns has to be converted to appropriate format.
Variables that contain label values instead of numeric values are known as categorical data.

Problem with Categorical Data:

Many machine learning algorithms are not capable of operating on label data directly. All input and output variables are required to be numeric. This is a constraint for efficient implementation of machine learning algorithms rather than hard limitations on the algorithms themselves. This means that categorical data must be converted to a numerical form, so a machine learning algorithm can process it. Methods to convert:

i. One-hot encoding
ii. Response encoding (Mean Imputation)

Evaluating the columns:
The columns were evaluated to check whether they are impactful for predicting the Class column which is the dependent variable.

Gene Column:

From the Gene column, it was observed that the first 50 unique genes, contributed m almost 75% of the total values. The Laplace Smoothing was performed on the Gene column and it was evaluated that the train, cross-validation, and test

log-loss were approximately 1.04, 1.20, 1.20 for alpha = 0.0001. Also, the overlap between train, test, and train, cross-validation was found to be 95.78% and 96.05% respectively. It was therefore observed that Gene column is good prediction variable as it has very low log- loss value compared to Random Model and high stability as the overlap is high.

```
print('Number of Unique Genes :', unique_genes.shape[0])

# Top 10 genes
print(unique_genes.head(10))

Number of Unique Genes : 235
BRCA1      173
TP53       103
EGFR        87
PTEN        79
BRCA2       79
BRAF        66
KIT         62
ERBB2       47
ALK         42
PIK3CA      41
```

Variation Column:

From the Variation column, it was observed that first 1500 unique variations, contributes almost 80% of the total values. The Laplace Smoothing was performed on the Variation column and it was evaluated that the train, cross-validation, and test log-loss were approximately 1.05, 1.69, 1.74 for alpha = 0.01. Also, the overlap between train, test, and train, cross-validation was found to be 7.51% and 12.40% respectively. It was therefore observed that Variation column is good prediction variable as it has very low log-loss value compared to Random Model even though the stability is low due to low overlap.

```
print('Number of Unique Variations :', unique_variations.shape[0])
# the top 10 variations that occured most
print(unique_variations.head(10))

Number of Unique Variations : 1921
Truncating_Mutations    64
Amplification           48
Deletion                46
Fusions                 23
Overexpression           5
G12V                     4
E330K                    2
M1R                      2
T73I                     2
Q22K                     2
```

Text Column:

For the Text column, the unique words were stored in a dictionary and the corresponding count value is incremented every time the word is encountered. A count vectorizer was built with all the words that occurred a minimum three times in the data. The total unique words were found to be 52944. Using Response Encoding values in each row were converted such that the sum was one. Every feature is normalized and is trained using the same Vectorizer as the train data. It was evaluated that the train, cross-validation, and test log-loss were approximately 0.77, 1.20, 1.19 for alpha = 0.001. Also, the overlap between train, test, and train, cross-validation was found to be 96.51% and 97.59% respectively. It was therefore observed that the Text column is good prediction variable as it has very low log-loss value compared to Random Model and high stability as the overlap is high.

3.2.9. Data Preparation

For the successful generation of the log-loss confusion matrix, precision matrix, recall matrix and determining the misclassified points functions were prepared in advance.

3.2.10. Combining the Features

All the three variables as generated using One-hot Encoding and Response Encoding were combined together respectively using numpy function hstack.

According to one-hot encoding the features are:
1. Train (2124, 55129)
2. Test (665, 55129)
3. Cross-Validation (532, 55129)

According to Response encoding the features are:
1. Train (2124, 27)
2. Test (665, 27)
3. Cross-Validation (532, 27)

Hence, we can deduce that Response Encoding would limit the number of columns in the best possible way.

3.2.11. Building Machine Learning Model

Naïve Bayes

Naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

They are highly scalable, which requires a number of parameters linear in the number of variables in a learning problem. Training can be done by evaluating a closed-form expression, which takes linear time.

For classification with discrete features (e.g., word counts for text classification), the multinomial Naive Bayes classifier is suitable. It normally requires integer feature counts.

The Multinomial Naïve Bayes classifier was used in this project as we have discrete data in the form of 9 classes and also because the Naïve Bayes model is highly interpretable.

K-Nearest N

K-Nearest Neighbors algorithm (KNN) is a non- parametric method used for classification and regression. In both cases, the input contains k closest training examples in the feature space. Based on whether the KNN is used for classification or regression, the output changes.

i. In KNN classification, the output is a class membership. Based on a plurality vote of its neighbors, an object is classified with the object being assigned to the class most common among its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of that single nearest neighbor.

ii. In KNN regression, the output is the property value for the object. The resulting value is the average of the values of k nearest neighbors.

Logistic Regression

It is a Machine Learning classification algorithm which is used to predict the probability of a categorical dependent variable. In logistic regression, the dependent variable is a binary variable that contains data coded as 1 (yes, success, etc.) or 0 (no, failure, etc.). The logistic regression model predicts P(Y=1) as a function of X.

The Logistic Regression model was used because it is highly interpretable and can be used for Multi-class classification easily.

For the LR model we would use two methods:

i. Over-sampling which means classes with fewer data would be balanced out with respect to other classes.
ii. Without balancing which means classes won't be balanced out.

Linear Support Vector Machine

Support Vector Machines are based on the idea of finding a hyperplane that best divides a dataset into two classes. Data points which are nearest to the hyperplane, are Support vectors. Because of which, they are considered as the critical elements of a data set.

The Linear Support Vector Machine was used because it is a highly interpretable model.

Random Forest Classifier

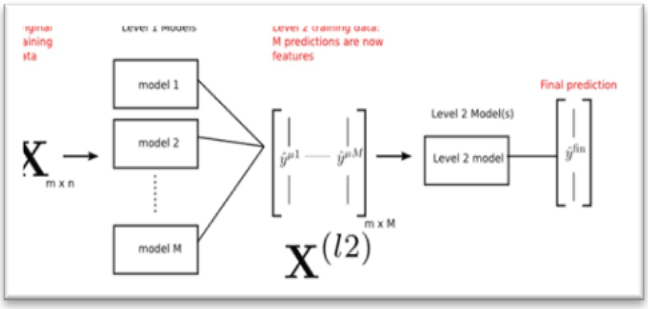
Random forests is a supervised learning algorithm, which can be used for both classification and regression. It is also the most flexible and easy to use the algorithm. A forest is comprised of trees. The more trees it has, the more robust a forest is. It creates decision trees on randomly selected data samples, get a prediction from each tree and selects the best solution by means of voting. Random Forest Classifier is an ensemble algorithm. Ensemble algorithms are those which combines more than one algorithms of a same or different kind for classifying objects. For example, running prediction over Naive Bayes, SVM, and Decision Tree and then taking a vote for final consideration of class for a test object.

The important term Bagging directs to the ensemble, which means that a group of thing viewed as a whole. In order to ensure the ensemble, we need to do the following:

i. We should create multiple models
ii. We should combine their results.

Stacking Model

Stacked Generalization or stacking is an ensemble algorithm where a new model is trained to combine the predictions from two or more models already trained or your dataset. The predictions from the existing models or submodels are combined using a new model, and as such stacking is often referred to as blending, as the predictions from sub-models are blended together. It is typical to use a simple linear method to combine the predictions for submodels such as simple averaging or voting, to a weighted sum using linear regression or logistic regression. Models that have their predictions combined must have skill on the problem, but do not need to be the best possible models. This means as long as the model shows some advantage over a baseline prediction, you do not need to tune the submodels intently.



The stacking classifier produces the train, cross-validation, and tests log-loss values as 0.67, 1.18, 1.16 respectively. The misclassified points were 38.64% which means that the model predicts 61.36% points correctly. The confusion, precision and recall matrix support these results.

Maximum Voting Classifier

One of the simplest ways of combining predictions from multiple machine learning algorithms is by voting. It first creates two or more standalone models from the training dataset. It can then be used to wrap the models and average the predictions of the sub-models when asked to make predictions for new data. A voting ensemble model for classification can be created using the Voting Classifier class.

The maximum voting classifier uses Logistic Regression, Linear Support Vector Machine, and Random Forest Classifier and produces the train, cross-validation, and test log-loss values as 0.92, 1.22, 1.22 respectively. The misclassified points were 37.29% which means that the model predicts 62.71% points correctly. The confusion, precision and recall matrix support these results.

3.2.12. Working of the System:

Frontend:

i. The user will input the genetic variation and gene in the system.
ii. As the user clicks the Prediction Button on the system these variation and gene are carried to the backend where they are processed to make an appropriate class prediction.

Backend:

i. At the backend side, all the necessary libraries including pandas, numpy, nltk, gensim, etc. are imported.
ii. After importing all the libraries, the first step is Data Preprocessing where the user input is processed by removing numbers, inappropriate spaces and converting it into the lower case to avoid errors.
iii. After Data pre-processing, the user data is cleaned and the missing values present in the data are handled by replacing the null values with the concatenation of GENE and VARIATION column. The data were cross-checked for null values after imputation.
iv. In the next step the LabelEncoder is used which encodes categorical features as a one-hot numeric array. LabelEncoder is used to normalize labels. LabelEncoder transforms non-numerical user data to numerical labels. This is done in order to ensure that the user data is accepted by the model as the model can only process numerical data.
v. This user data is then passed to the previously trained and saved model. The model then makes the class prediction for the user input.
vi. This predicted class is then displayed on the frontend.

4. RESULTS

The result represents the classification of all the ID's in the dataset into their corresponding classes.

	ID	class1	class2	class3	class4	class5	class6	class7	class8	class9
0	0	0.656126	0.124289	0.001102	0.095194	0.001501	0.001743	0.119688	0.000124	0.000233
1	1	0.083554	0.730913	0.001339	0.148635	0.002058	0.002459	0.030564	0.000159	0.000318
2	2	0.083554	0.730913	0.001339	0.148635	0.002058	0.002459	0.030564	0.000159	0.000318
3	3	0.194547	0.076416	0.023644	0.553465	0.001286	0.001474	0.148863	0.000104	0.000202
4	4	0.000020	0.000020	0.000000	0.623926	0.375771	0.000263	0.000000	0.000000	0.000000
...	...	...	...	...	...	...	...	...	...	...
95	95	0.194547	0.076416	0.023644	0.553465	0.001286	0.001474	0.148863	0.000104	0.000202
96	96	0.070832	0.021223	0.001069	0.616794	0.002782	0.003307	0.283347	0.000216	0.000431
97	97	0.194547	0.076416	0.023644	0.553465	0.001286	0.001474	0.148863	0.000104	0.000202
98	98	0.194547	0.076416	0.023644	0.553465	0.001286	0.001474	0.148863	0.000104	0.000202
99	99	0.168213	0.137239	0.027258	0.199420	0.074107	0.084920	0.291465	0.005889	0.011488

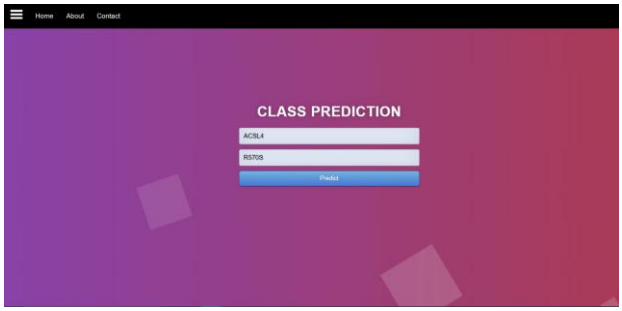
100 rows × 10 columns

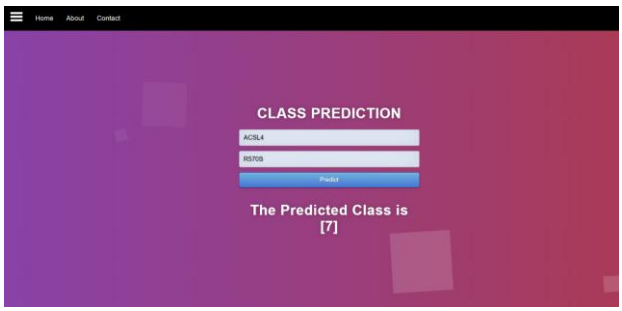
5. DISCUSSION

Comparison of different models used:

Model	Train log-loss	Cross-Validation log-loss	Test log-loss	Miss-classified %	Log-loss
Naïve-Bayes	0.9	1.18	1.27	34.58	1.18
K Nearest Neighbors	0.60	0.95	1.03	31.95	0.95
Logistic Regression (With balancing)	0.54	1.02	1.05	29.69	1.02
Logistic Regression (Without balancing)	0.53	1.04	1.06	29.67	1.04
Linear Support Vector Machine	0.75	1.07	1.11	31.1	1.074
Random Forest Classifier (One-hot encoding)	0.67	1.13	1.15	36.84	1.12
Random Forest Classifier (Response encoding)	0.05	1.23	1.31	41.72	1.23

System Screenshots:





6. CONCLUSION AND FUTURE WORK

A well-organized gene database promotes accurate phenotype-driven gene analysis. It enables classification as new studies are published, providing patients with a diagnosis and opportunities for therapeutic options, and an end to their "diagnostic journey." This classification system is simple enough to be quickly implemented, and it can specifically guide reporting decisions at the important boundary of limited and moderate evidence, determining whether a gene is characterized.

The minimal value for the Log-Loss is obtained on implementing K Nearest Neighbors algorithm on the gene, variation, and text data value points. Although the Log-Loss for K Nearest Neighbors algorithm is less, we choose Logistic Regression (With balancing) because KNN gives an overfitted value (1.008).

As a future scope, the similar gene mutation and classification techniques can be extended to find a cure to diseases other than Cancer and can be a breakthrough technology in personalized medical space.

REFERENCES

[1] Technologies for deriving primary tumour cells for use in personalized cancer therapy Abhisek Mitra, Lopa Mishra, Shulin Li.

[2] Breast Cancer Prediction and Detection using Data Mining Classification Algorithms a Comparative Study.

[3] Yi-Sheng Sun, Zhao Zhao, Han-Ping-Zhu,"Risk factors and Preventions of Breast Cancer" International Journal of Biological Sciences.

[4] AlirezaOsarech, BitaShadgar,"A Computer Aided Diagnosis System for Breast Cancer",International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011

[5] https://www.kaggle.com/c/msk-redefining-cancer-treatment