ISSN No:-2456-2165

Classification of Genetic Mutation using Machine Learning

Karan A. Solanki
Dept. Information Technology
Bharati Vidyapeeth College of Engineering,
Belpada, Navi Mumbai, India

Sonali A. Shinde
Dept. Information Technology
Bharati Vidyapeeth College of Engineering,
Belpada, Navi Mumbai, India

Anish N. Shelte
Dept. Information Technology
Bharati Vidyapeeth College of Engineering,
Belpada, Navi Mumbai, India

Abstract:- Genes are pieces of DNA (deoxyribonucleic acid) inside our cells that tell the cell how to make the proteins the body needs to function. DNA is the genetic "blueprint" in each cell. Genes are inherited from a parent to a child, such as hair color, eye color, and height. They can also affect whether a person is likely to develop some certain diseases, such as cancer. Changes in genes is called as mutations, that play an important role in the development of cancer. Certain mutations can cause cells to grow out of control, which can lead to a cancer. Most cancers start due to this acquired gene mutations that happens during a person's lifetime. Sometimes these gene changes have an outside cause or such as exposure to sun-light or tobacco. Molecular Pathologists, who are interested in a particular Gene, select that gene and a variation. The Molecular Pathologists then collect all the research work (Medical literature) that has ever been done on that particular gene and its variation. The Domain Expert then spends huge amount of text reading the text and then they finally determine which class this gene and variation belong to. This project proposes an efficient solution to automate the step where the Molecular Pathologist manually spends time reading the Medical Literature to determine if the gene and a particular variation on that gene belong to a certain class of cancer or not using Machine Learning.

Keywords:- *Data Analysis, Variation, Mutation, Machine Learning, Feature Extraction, Feature Selection, classification. Genetic Testing, Natural Language Programming.*

I. INTRODUCTION

A gene mutation may be a permanent alteration with in the DNA sequence that creates up a gene, such the sequence differs from what is found in most of the people. Mutations range in size; they will affect anywhere from one DNA building block (base pair) to an outsized segment of a chromosome that has multiple genes. Genes are pieces of DNA (deoxyribonucleic acid) inside our cells that tell the cell the way to make the proteins the body must function. DNA is that the genetic "blueprint" in each cell. Genes affect inherited traits passed on from a parent to a toddler, like hair color, eye color, and height. they will also affect whether an individual is probably going to develop certain diseases, like cancer. Changes in genes, called mutations, play a crucial role within the development of the cancer might be defined the health care industry today isn't what it had been just five years ago. this is often largely thanks to technology and an outsized number of innovative digital solutions that are introduced a day. Many technological solutions are proposed for several problems that the planet of drugs was facing, and these have greatly changed and improved the drugs industry. There are many breakthroughs in the data collection, research, treatments, and medical devices like hearing aids which have had an enormous impact on the planet of the medicines. Today, because of technology, there's better and more accessible treatment for a good sort of diseases, better and more efficient look after the sick and better health care and disease control. a number of technologies like Machine learning had a big impact in many areas of science and technology, that including bioscience and medical research. during this we highlight the most recent advances that are made within the development of machine learning algorithms for different fundamental aspects like statistical bioinformatics to their deployment in clinical diagnosis, prognosis and drug development. Our project proposes an efficient solution to automate the step where the Molecular Pathologist manually spends time reading the Medical Literature to work out if the gene and a specific variation thereon gene belong to a particular class of cancer or not using Machine Learning.

Machine Learning (ML) is one among the core branches of AI . It's a system which takes in data, finds patterns, trains itself using the info and outputs an outcome. Machines can do something good which humans are not that good at. they will repeat themselves thousands of times without getting exhausted. After every iteration, the machine repeats the method to try to to it better. Humans roll in the hay too; we call it practice. While practice may make perfect, no amount of practice can put a person's even on the brink of the computational speed of a Computer. A biopsy usually takes a Pathologist 10 day. A computer can do thousands of biopsies during a matter of seconds. Another advantage is that the great accuracy of machines.

ISSN No:-2456-2165

With the arrival of the web of Things technology, there's such a lot data call at the planet that humans can't possibly undergo it all. That's where machines help us.

A. *Problem Statement*

Researchers face lots of problems when they have to classify a particular gene, they manually refer books and research papers. Hence, we are developing a Machine Learning Classifier that can automate the process of classification of genes based on the medical evidence. A molecular pathologist selects an inventory of genetic variations. The molecular pathologist search for evidence with in the medical literature those somehow are relevant to the genetic variations of interest. Finally, this molecular pathologist spends an enormous amount of your time analyzing the evidence associated with each of the variations to classify them.

B. *Scope*

Diagnosis via machine learning works when the condition is often reduced to classification task, in areas where we currently believe the clinician to be ready to visually identify patterns that indicate the presence or sort of the condition. Hence, the Machine Learning can inflate all the attempt traditionally and left only to pathologists with microscopes. Understanding the genetic mutations that basically matter during a cancer tumor may be a really challenging task with a possible huge impact on many lives.

C. *Objectives*

- Study of Existing System
- Preparing Comparison of Existing System.
- Preparing System architecture.
- Study of System Design.
- Prepare Project organization.
- Prepare Project Gantt chart.
- Do Implementation and provide result.

II. EXISTING SYSTEM

A molecular pathologist selects a list of genetic mutations from the patient which may have variations. The molecular pathologist then searches for evidence in the medical literature that somehow is relevant to the genetic mutation of interest. Finally, this molecular pathologist spends an enormous amount of your time analyzing the evidence associated with each of the variations to classify the category .This take at least a week to create a report and send it to the patient about which class of mutation she is suffering so the further treatment accordingly could be taken by the patient.

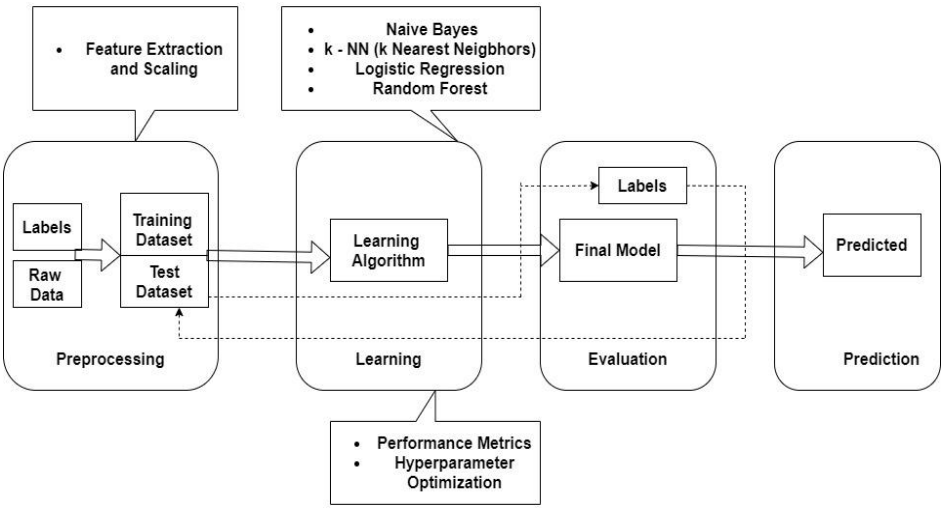

Fig 1:- System Architecture

A molecular pathologist can select a dataset, and perform many operations on that database. There are various steps followed. The first step is exploratory data analysis (EDA). Preprocessing of data include the stop words removal. This step includes preprocessing of the medical literature text. Stop words are a group of commonly used words during a language. samples of stop words in English are "a", "the", "is", "are" and etc. The intuition behind using stop words is that, by removing irrelevant information from medical literature, we will instead focus on important words clinical and medical words instead. The dataset is then split into Test and Training sets. Encoding techniques such as Bag of Words and TF IDF are applied. Once we have the encoded text, it is appended with encoded (using One Hot Encoding) categorical variable such as Gene and Variation. Models are then trained using Machine Learning Algorithms such as Naïve Bays, Logistic Regression, Random Forest and k-NN.

III. SYSTEM DESIGN

System design is the process of defining the architecture, components, and modules, how they are finding variation and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

A. Implementation Details

The Implementation Process are as follows :

➢ *EDA*

In statistics, exploratory data analysis (EDA) is an approach for analyzing data sets that summarize their main characteristics, often with visual methods. Primarily EDA is for seeing what information can be derived beyond the formal modeling or hypothesis. EDA is different from initial data analysis (IDA) which focuses more narrowly on checking assumptions required for model fitting and hypothesis testing, and handling missing values and making transformations of variables as required. EDA encompasses IDA.

➢ *Preprocessing of medical text (stop words removal)*

To preprocess your text simply means to bring your text into a form that's predictable and analyzable for your task. a specific task may be a combination of approach and domain. There are many various ways to preprocess your text.

- *Stop-word removal*

Stop words are a group of commonly used words during a language. samples of stop words in English are "a", "the", "is", "are" and etc. The intuition behind using stop words is that, by removing irrelevant information from medical literature, we will instead focus on important clinical and medical words instead, within the context of an enquiry system.

➢ *Encoding of medical text:*

Bag of Words: The bag-of-words model may be a simplifying representation utilized in tongue processing and knowledge retrieval (IR). during this model, a text (such as a sentence or a document) is represented because the bag (multiset) of its words, disregarding grammar and even ordering but keeping multiplicity. The bag-of-words model has also been used for computer vision.

Intelligent applications creates intelligent business processes

intelligent	applications	creates	business	processes
2	1	1	1	1

Fig 2:- Bag of words

The bag-of-words model is usually utilized in methods of document classification where the (frequency of) occurrence of every word is employed as a feature for training a classifier The Bag-of-words model is especially used as a tool of feature generation. After transforming the text into a "bag of words", we will calculate various measures to characterize the text. the foremost common sort of characteristics, or features calculated from the Bag-of-words model is term frequency, namely, the amount of times a term appears within the text.

TF IDF: TF*IDF is an information retrieval technique that weighs a term's frequency (TF) and its inverse document frequency (IDF). Each word or term has its respective TF and IDF score. the merchandise of the TF and IDF many a term is named the TF*IDF weight of that term. Put simply, the upper the TF*IDF score (weight), the rarer the term and the other way around. The TF*IDF algorithm is employed to weigh a keyword in any content and assign the importance thereto keyword supported the amount of times it appears within the document. More importantly, it checks how relevant the keyword is throughout the online, which is mentioned as corpus.

sentence 1	earth is the third planet from the sun
sentence 2	Jupiter is the largest planet

Word	TF (Sentence 1)	TF (Sentence 2)	IDF	TF*IDF (sentence 1)	TF*IDF (Sentence 2)
earth	1/8	0	log(2/1)=0	0.0375	0
is	1/8	1/5	log(2/2)=0	0	0
the	2/8	1/5	log(2/2)=0	0	0
third	1/8	0	log(2/1)=0.3	0.0375	0
planet	1/8	1/5	log(2/2)=0	0	0
from	0	0	log(2/1)=0.3	0	0
sun	1/8	0	log(2/1)=0.3	0.0375	0
largest	0	1/5	log(2/1)=0.3	0	0.06
Jupiter	0	1/5	log(2/1)=0.3	0	0.06

Fig 3:- TF IDF

For a term t during a document d, the load Wt,d of term t in document d is given by:

Wt,d = TFt,d log (N/DFt)

Where,
* TFt,d is that the number of occurrences of t in document d.
* DFt is that the number of documents containing the term t.
* N is that the total number of documents within the corpus.

➢ *Encoding of Categorical data:*

Categorical Encoding refers to reworking a categorical feature into one or multiple numeric features. you'll use any mathematical method or logical method you would like to rework the specific feature. Categorical Data is that the data that generally takes a limited number of possible values. All machine learning models are some quite mathematical model that needs numbers to figure with. this is often one among the first reasons we'd like to pre-process the specific data before we will feed it to machine learning models.

➢ *Split into train and test dataset:*

Dataset is split int 80:20 ratio, 80 for training dataset and 20 for testing dataset.

➢ *For Training dataset, we used the following algorithms to check which algorithm (along with a text encoding technique) gives the lowest log-loss:*

We used the following algorithms.
- Naive Bayes
- Logistic Regression
- Random Forest
- k-NN

We also used CalibratedClassifierCV to get calibrated probability values.

- *Testing (Metric Log-loss)*

Log Loss is that the most vital classification metric supported probabilities.

Interpret raw log-loss values, but log-loss remains an honest metric for comparing models. For any given problem, a lower log-loss value means better predictions.

So as to calculate Log Loss the classifier must assign a probability to every class instead of simply yielding the foremost likely class. Mathematically Log Loss is defined as

$$-\frac{1}{N}\Sigma\sum_{i=1}^{N}\Sigma_{j=1}^{M}\,yijlogPij$$

where N is that the number of samples or instances, M is that the number of possible labels, yijyij may be a binary indicator of whether or not label j is that the correct classification as an example i, and pijpij is that the model probability of assigning label j to instance i. an ideal classifier would have a Log Loss of precisely zero. Less ideal classifiers have progressively larger values of Log Loss. If there are only two classes then the expression above simplifies to

$$-\frac{1}{N}\sum_{i=1}^{N}[yilog\ Pi + (1-yi)\log(1-pi)]$$

- *Deploying a model (which has the minimum log-loss) locally using flask:*

IV. IMPLEMENTATION OUTPUTS

A. Result Analysis

Comparison of the log-loss obtained during training. See Fig-11. Once after getting the best hyper-parameter (with the lowest log-loss), we trained a new model again with that best hyper-parameter obtained. The chosen log-loss value, for training and testing, is the minimum of all the values obtained after training the respective model with different hyper-parameters of algorithms. For example:

- We choose the best 'alpha' value, the one giving us the lowest log-loss in case of Naïve Bayes.
- We choose the best 'K' value, the one giving us the lowest log-loss in case of k-NN.
- We choose the best number of 'estimators' value, the one giving us the lowest log-loss in case of Random Forest.
- We choose the best 'Inverse Regularization' value, the one giving us the lowest log-loss in case of Logistic Regression.

Consider the subsequent. For Bag of Words encoding of Naïve Bayes, the minimum log-loss obtained is 1.26 for alpha value 0.00001.

```
In [58]: %%time

         from sklearn.naive_bayes import MultinomialNB
         from sklearn.calibration import CalibratedClassifierCV
         from sklearn.metrics.classification import accuracy_score, log_loss

         alpha = [0.00001, 0.0001, 0.001, 0.1, 1, 10, 100,1000]

         test_log_error_array = []

         print('***Logloss while training with Naive Bayes*** \n')
         print('*** Set 1 (Bag-of-Words Encoding of Medical Text)*** \n')

         for i in alpha:
             print("For alpha =", i)
             model = MultinomialNB(alpha=i)
             model.fit(X_train_bow, y_train)

             sig_clf = CalibratedClassifierCV(model, method="sigmoid")
             sig_clf.fit(X_train_bow, y_train)

             sig_clf_probs = sig_clf.predict_proba(X_test_bow)

             test_log_error_array.append(log_loss(y_test, sig_clf_probs, labels=model.classes_))

             print("Log Loss :", log_loss(y_test, sig_clf_probs))
```

Fig 4:- Training the matrix with different hyper- parameter

```
In [81]: %%time

         best_alpha = 0.00001

         clf = MultinomialNB(alpha=best_alpha)
         clf.fit(X_train_bow, y_train)
         sig_clf_test = CalibratedClassifierCV(clf, method="sigmoid")

         sig_clf_test.fit(X_train_bow, y_train)
         sig_clf_probs = sig_clf.predict_proba(X_test_bow)

         print('***Logloss while testing with Naive Bayes*** \n')
         print('*** Set 1 (Bag-of-Words Encoding of Medical Text)*** \n')
         print("Log Loss :",log_loss(y_test, sig_clf_probs))
```

Fig 5:- Testing the matrix with the best hyper-parameter

```
***Logloss while training with Naive Bayes***

*** Set 1 (Bag-of-Words Encoding of Medical Text)***

For alpha = 1e-05
Log Loss : 1.2636815276040476
For alpha = 0.0001
Log Loss : 1.264328859870972
For alpha = 0.001
Log Loss : 1.2733282236016814
For alpha = 0.1
Log Loss : 1.2923463745174795
For alpha = 1
Log Loss : 1.291511406209013
For alpha = 10
Log Loss : 1.3155539560903213
For alpha = 100
Log Loss : 1.630899190698445
For alpha = 1000
Log Loss : 1.770077966676224
Wall time: 16.7 s
```

Fig 6:- Log-loss obtained while training using Naïve Bayes

```
***Logloss while testing with Naive Bayes***

*** Set 1 (Bag-of-Words Encoding of Medical Text)***

Log Loss : 1.770077966676224
Wall time: 1.87 s
```

Fig 7:- Log-loss obtained while testing using Naïve Bayes

ISSN No:-2456-2165

```
*** Set 1 of Naive Bayes (Bag-of-Words Encoding of Medical Text)***

***Analyzing the acutal and predicted class by taking a test datapoint***

Predicted Class : Gain-of-function
Predicted Class Probabilities: [0.61522791 0.03814413 0.0837725  0.08099966 0.03923457 0.00544302
 0.12026088 0.01419579 0.00272154]
Actual Class : Gain-of-function
**************************************************
Gain-of-function 0.615227906815719
Inconclusive 0.038144138038824605
Likely Gain-of-function 0.0837724962346252
Likely Loss-of-function 0.08099966344887025
Likely Neutral 0.03923457215181567
Likely Switch-of-function 0.005443020782834237
Loss-of-function 0.1202608764220242
Neutral 0.014195790219329055
Switch-of-function 0.0027215435365362203
```

Fig 8:- Analysing a sample data point

```
Classification Report for Naive Bayes

*** Set 1 (Bag-of-Words Encoding of Medical Text)***

                        precision    recall  f1-score   support

          Gain-of-function       0.69      0.76      0.72       207
              Inconclusive       0.79      0.56      0.65        54
    Likely Gain-of-function       0.54      0.65      0.59        93
    Likely Loss-of-function       0.52      0.62      0.56       107
             Likely Neutral       0.45      0.46      0.46        52
    Likely Switch-of-function       0.00      0.00      0.00         2
          Loss-of-function       0.79      0.49      0.60       123
                   Neutral       0.55      0.63      0.59        19
        Switch-of-function       0.60      0.86      0.71         7

                  accuracy                           0.62       664
                 macro avg       0.55      0.56      0.54       664
              weighted avg       0.64      0.62      0.62       664
```

Fig 9:- Classification report for Naïve Bayes

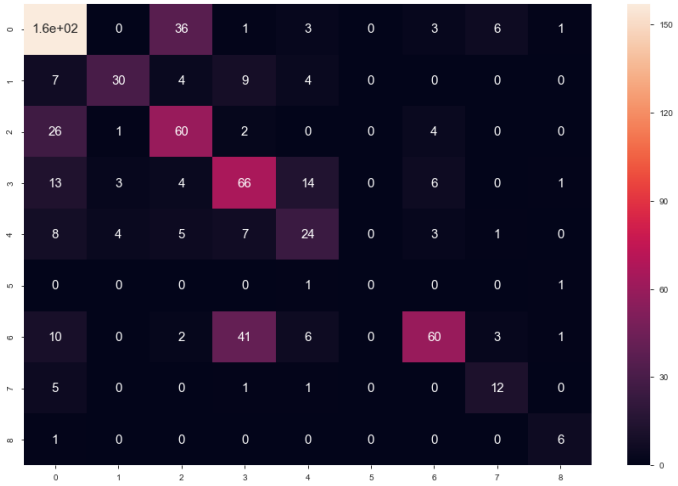

Fig 10:- Confusion Matrix for Naïve Bayes

Algorithm	Encoding	Minimum log-loss while training	Log-loss obtained after testing
Naive Bayes	Bag of Words	1.26	1.77
Naive Bayes	TF-IDF	1.11	1.24
k-NN (Nearest Neighbors)	Bag of Words	1.21	2.89
k-NN (Nearest Neighbors)	TF-IDF	1.12	2.76
Logistic Regression	Bag of Words	1.15	1.37
Logistic Regression	TF-IDF	0.9	0.89
Random Forest	Bag of Words	1.21	1.71
Random Forest	TF-IDF	1.15	1.72

Fig 11:- Log-loss obtained while testing using Naïve Bayes

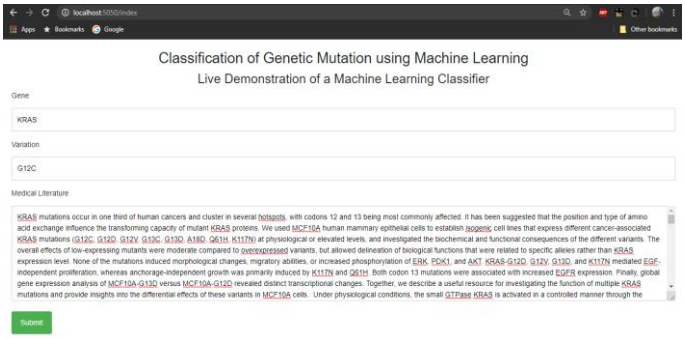

Fig 12:- Web Page after successfully running the Flask API.

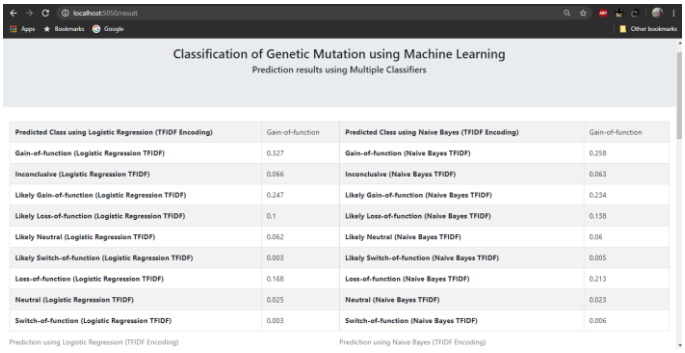

Fig 13:- Output tables after clicking the submit button, it shows predicted class-label and probabilities of all classes.

V. CONCLUSION

Gene mutation in general causes genetic changes which can lead to diseases. To achieve our goal of minimizing the part where a pathologist manually spends more time, by using various Machine Learning algorithm's such as Naïve Bayes, Logistic Regression, Random forest and k-NN.

We used the following eight models to determine the mutation, given a gene, variation and the medical literature published.

- Logistic Regression (with TFIDF Encoding): Logistic Regression with TFIDF encoding was the best model of all eight models with a log-loss value of 0.9 while training, and 0.89 after testing. The accuracy was this model was also the highest among all eight models, that is, 68%.
- Naïve Bayes (with TFIDF Encoding): Naïve Bayes with TFIDF encoding performed slightly less desirable than Logistic Regression with TFIDF encoding. It performed with a log-loss value of 1.11 while training and 1.24 after testing. Accuracy was 65%.
- Logistic Regression (with Bag of Words Encoding): It performed with a log-loss value of 1.15 while training and 1.37 after testing. Accuracy of this classifier was 62%.
- Naïve Bayes (with Bag of Words Encoding): It performed with a log-loss value of 1.26 while training and 1.77 after testing. Accuracy of this model was 63%.

- Random Forest (with TFIDF Encoding): It performed with a log-loss value of 1.15 while training and 1.72 after testing. Accuracy of this classifier was 62%.
- K-NN (with TFIDF Encoding): It performed with a log-loss value of 1.12 while training and 2.76 after testing. Accuracy of this classifier was 62%.
- Random Forest (with Bag of Words Encoding): It performed with a log-loss value of 1.21 while training and 1.71 after testing. Accuracy of this classifier was 61%.
- K-NN (with Bag of Words Encoding): K-NN with Bag of Words encoding was the worst performing model of all eight models with a log-loss value of 1.21 while training, and 2.89 after testing. The accuracy was this model was the lowest among all eight models, that is, 58%.

Therefore, the conclusion is that the proposed models can be used to classify cancerous mutations in a clinical setting. Apart from a decent classification accuracy achieved, there are still directions for further improvement of the proposed model. The project is not just limited to identifying and classifying cancerous mutations. With a well-defined and error free data we can also use such classifiers for other multi-factorial genetic disorders.

ACKNOWLEDGEMENT

Commencing with our final year project "Classification of Genetic Mutation using Machine Learning " which might be decider of all the efforts taken throughout these 4 years was a touch hesitant. But this dilemma was soon put to stake by the type of support we received throughout the building of our idea into work.

Our project guide Prof. H. A. Chavan has been an interesting support throughout Working under his guidance made our work simple as he always welcomed us with the silliest doubts, we had with a curiosity to assist us and widen our knowledge. Having given the privilege to figure under such cooperative and helpful faculty boosted our confidence to excel in every task we took up during the working of our project.

We would wish to thank our Project coordinator Prof. Sonali Mane who kept on guiding us and informing us about the schedule of the execution of our project. Without her help we guess, the various doubts we had in our minds would haven't been solved. We are grateful to our Head of Department Dr. S. M. Patil.

And last but not the smallest amount we might wish to express our sincere gratitude to respected Principal Madam Dr. S. D. Jadhav for allowing our little mind to think in several directions and help us broaden our horizons by making available all the required amenities needed within the working of our project. Also not forgetting our parents and friends who are supporting us and helping us altogether the possible ways they will.

REFERENCES

- *Journal Paper*

[1]. Multiple SVM-RFE for Gene Selection in Cancer Classification With Expression Data,Kai-Bo Duan, Jagath C. Rajapakse*, Senior Member, IEEE, Haiying Wang, Member, IEEE, and Francisco Azuaje, Senior Member, IEEE
[2]. Cancer Classification of Gene Expression Data using Machine Learning Models. Joseph M. De Guia School of IT Mapua University,658 Muralla St., Intramuros, Manila Philippines 1002 jmdeguia@mapua.edu.ph
[3]. Gene Classification Using Expression Profiles: A Feasibility Study ,Michihiro Kuramochi and George Karypis Department of Computer Science/Army HPC Research Center, University of Minnesota Minneapolis, MN 55455 fkuram, karypisg@cs.umn.edu

- *Proceeding paper*

[4]. Classification of Cancerous Profiles using Machine Learning**.** 22 March 2018. 2017 International Conference on Machine Learning and Data Science (MLDS).
[5]. https://ieeexplore.ieee.org/document/1501840
[6]. https://www.kaggle.com/c/msk-redefining-cancer-treatment
[7]. https://www.onely.com/blog/what-is-tf
[8]. https://ieeexplore.ieee.org/document/1501840
[9]. https://www.kaggle.com/c/msk-redefining-cancer-treatment
[10]. https://www.onely.com/blog/what-is-tf-idf/
[11]. https://datacarpentry.org/python-socialsci/11-joins/index.html
[12]. https://www.kaggle.com/c/msk-redefining-cancer-treatment/discussion/35810
[13]. https://towardsdatascience.com/a-better-eda-with-pandas-profiling-e842a00e1136
[14]. http://zetcode.com/python/prettytable/
[15]. https://stackoverflow.com/questions/1987694/how-to-print-the-full-numpy-array-without-truncation
[16]. https://chrisalbon.com/python/data_wrangling/pandas_list_unique_values_in_column/
[17]. https://stackoverflow.com/questions/55881651/how-to-print-a-big-dimension-of-confusion-matrix
[18]. https://pymotw.com/2/re/
[19]. https://www.geeksforgeeks.org/removing-stop-words-nltk-python/
[20]. https://stackoverflow.com/questions/14247586/python-pandas-how-to-select-rows-with-one-or-more-nulls-from-a- dataframe-without
[21]. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.CountVectorizer.html