Leukaemia Classification using Machine Learning and Genomics

Vinamra Khoria and Amit Kumar

Vellore Institute of Technology, Vellore

vinamra.khoria2019@vitstudent.ac.in, amit.kumar2019@vitstudent.ac.in

Abstract. The field of genomics is vast and innovation is happening at a rapid pace today. With the availability of lots of medical data and extensive research, the tools at our disposal are sharper than ever. One such tool that has quite a lot of untapped potential is Machine Learning. Machine Learning is the field of computer science that gives computers the ability to understand data and make decisions based on that understanding, in quite a similar way as we humans do. Machine learning has proven to be the next big thing in almost all industries today including medicine. The use of Machine Learning in the field of genomics however, is yet to reach its true momentum. With the help of Machine Learning, patterns in genetic data can be found that were oblivious to us earlier and these patterns can be very useful in making conclusions about diseases and disorders that are inherently genetic in nature.

Keywords: Leukemia classification, KNN, Machine Learning , PCA.

1. Introduction

Cancer treatment has been one of the most active areas of medical research for many decades now. One of the main challenges that cancer treatment poses today is targeting tumor specific therapy. This is essential to maximize the efficiency of treatment and to reduce the toxicity of treatment at the same time. Accurate cancer classification is thus central to advancing treatment today. Till date, classification has mostly been done by observing the morphological appearance of tumors, but this approach is quite naive as different classes can often look similar, but react very differently to therapy. This calls for the need of a new approach for classifying cancer. This is where Genomics and Machine Learning come into the scenario. Gene expression data using DNA microarrays has been suggested to be able to provide a tool for classifying cancer. This is what we have set out to do in this chapter. We will be using gene expression data to build a class predictor taking acute leukemias as our test case. Leukemia has mainly two classes - acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Leukemia classification is a lengthy process with steps involved such as – interpreting the tumor's morphology, histochemistry, immunophenotyping, and cytogenetic analysis. All of these steps have to be carried out in separate, highly specialized laboratories. And although the classification is mostly accurate, errors still happen. In this chapter, we shall be using Machine Learning on gene expression data to try to build a classifier that correctly classifies ALL from AML.

2. Background

During the initial state of the project, a literature survey was done to get an understanding of KNN and data preprocessing. Various papers were reviewed to identify the performance factor to be analyzed. In the beginning, clustering and various different classification models were considered. But KNN was doing the most needful job. This has been considered by focussing the accuracy and

need of the problem statement. This literature review helped in the understanding of the methods for measuring and comparing performances of different models. It also helped to get the understanding of data preprocessing, different data exploration , better understanding of the problem statement and the industry. The deciding factor for the project has been accuracy of the model which as we will see is best achieved by KNN algorithm.

AUTHOR	CONTRIBUTION
1. **T. R. Golub**[1] 2. **D. K. Slonim,** 3. **P. Tamayo,** 4. **C. Huard,** 5. **M. Gaasenbeek,** 6. **J. P. Mesirov,** 7. **H. Coller,** 8. **M. L. Loh,** 9. **J. R. Downing,** 10. **M. A. Caligiuri,** 11. **C. D. Bloomfield,** 12. **E. S. Lander**	Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring
1. **T Haferlach** 2. **C Schoch** 3. **W. Hiddemann**	The New WHO Classification for Acute Myeloid Leukemias and Myelodysplastic Syndromes

3. Proposed model

3.1 K-Nearest Neighbors:

K Nearest Neighbors or KNN is an algorithm which can be used both for classification and regression purposes although it is more widely used for classification problems. To explain how KNN works, let us take an example of a classification problem. There are two classes in our example, one represented by red circles and the other by green squares. We have a point whose class is unknown represented by the blue star. To classify the unknown point correctly , we take a look at it's - 'k' nearest neighbors. Let k be 3 here. The 3 nearest points to our blue star are the red circles. Thus we can classify the point to belong to the class represented by the red circles. The more points of the same class that are included in the K nearest neighbors of the point we are analysing, the more confidently we can classify our point.

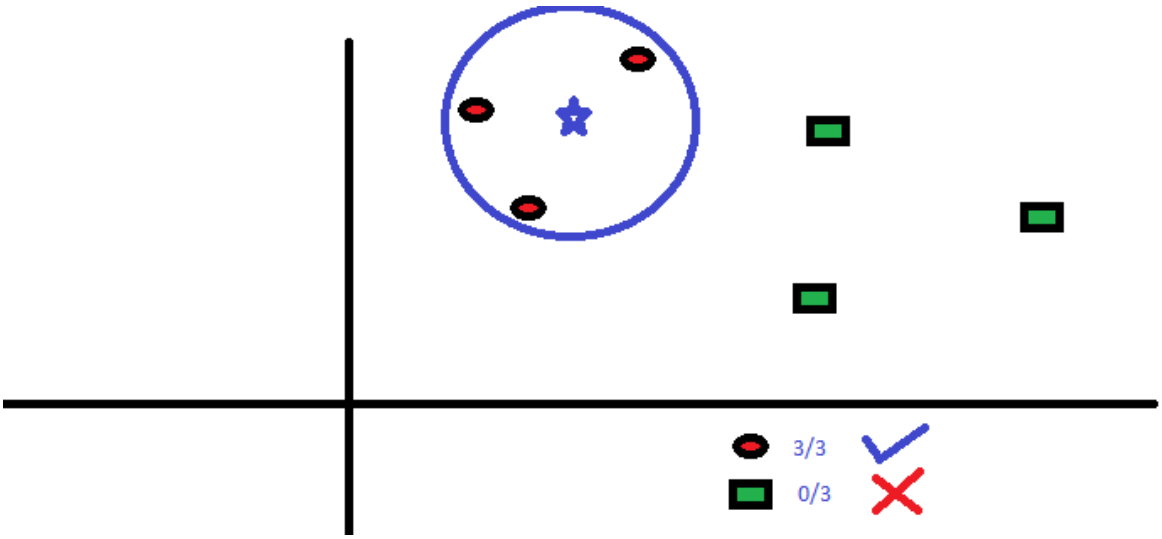

Image Source : analyticsvidhya.com

4. Experimental Results:

4.1 Dataset:

The dataset that we have used in the chapter comes from a proof-of-concept study published in 1999 by Golub et al. This data was used to prove that gene expression monitoring via DNA microarrays could classify new cases of cancer and hence provides a general approach for identifying new cancer classes and assigning tumors to known classes. There are two cancer classes namely acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL) into which the dataset was used to classify leukemia patients.

The data is broadly divided into **three sets**- the class labelings for all the patients, the training data which consists of 38 samples and the testing data which consists of 34 independent samples. We will use the 38 training samples to train our machine learning algorithm and the rest of the 34 samples to test its performance. Now let us look at the data more closely to understand what knowledge exactly will our classifier be learning .

4.2 Data Exploration:

4.2.1 First we look at the class labeling of our 72 patients which will be referred to by our classifier during the learning process.

Serial Number	Patient Id	Type of cancer
0	1	ALL
1	2	ALL

2	3	AML
3	4	AML
4	5	ALL

Table. 1. Classification data of leukemia patients into two categories - ALL and AML.

As we can see, a single row of the data contains a serial number, a patient ID, and the type of cancer the patient is suffering from. This is just a small part of the total 72 samples present in the data.

4.2.2 Second we look at the training data which contains the gene expression information for 38 patients from patient ID 1 to 38.

	Gene Description	Gene Accession Number	1	Call	2	call.1	3	call.2	4	call.3	5	call.4	6	call.5	7	call.6
0	AFFX-BioB-5_at (endogenous control)	AFFX-BioB-5_at	-214	A	-139	A	-76	A	-135	A	-106	A	-138	A	-72	A
1	AFFX-BioB-M_at (endogenous control)	AFFX-BioB-M_at	-153	A	-73	A	--49	A	-114	A	-125	A	-85	A	-144	A
2	AFFX-BioB_3_at (endogenous control)	AFFX-BioB3_at	-58	A	-1	A	-307	A	-265	A	-76	A	215	A	238	A
3	AFFX-BioC_5_at (endogenus control)	AFFX-BioC-5_at	88	A	283	A	309	A	12	A	168	A	71	A	55	A

Table. 2. Training dataset containing gene expression data for 38 patients

Each row represents a gene and it's expression in all the 38 patients. The first column contains the gene description. The second column contains a Gene Accession Number which is basically a unique identifier for the gene. The successive columns represent the patients with their information regarding this particular gene. Each patient is represented by two columns. The first column for a patient contains values of expression for the particular gene. For example - patient 1 has a value of -214 for the first gene - AFFX-BioB-5_at and patient 2 has a value of -139 for the same gene. The second column for each patient is the call column. The call columns are a decision on whether that gene is present in the sample in the preceding column. With patient 1 in the train set again, AFFX-BioB-5_at (index 0) is Absent, hum_alu_at (index 18) is Present, and D29642_at (index 341) is Marginal, which means it's too close to call. There are a total of 7129 rows which means that 7129 genes have been taken into consideration for all the patients. Our classifier will try to find patterns in this gene expression data by cross referencing the patients to the labelling data which we saw in the earlier section. These genes may be further feature engineered as we'll see ahead.

4.2.3 Third we look at the independent test data which contains the gene expression information for 34 patients from patient ID 39 to 72.

	Gene Description	Gene Accession Number	39	Call	40	call.1	42	call.2	47	call.3	48	call.4	49	call.5	41	call.6
0	AFFX-BioB-5_at (endogenous control)	AFFX-BioB-5_at	-342	A	-87	A	22	A	-243	A	-130	A	-256	A	-62	A
1	AFFX-BioB-M_at (endogenous control)	AFFX-BioB-M_at	-200	A	-248	A	-153	A	-218	A	-177	A	-249	A	-23	A
2	AFFX-BioB_3_at (endogenous control)	AFFX-BioB3_at	41	A	2662	A	17	A	-163	A	-28	A	-410	A	-7	A
3	AFFX-BioC_5_at (endogenous control)	AFFX-BioC-5_at	328	A	295	A	276	A	182	A	266	A	24	A	142	A

Table. 3 . Test dataset containing gene expression data for 34 patients.

This data is very similar to the training data. There are all the 7129 genes that we used to train the model, expressed for the 34 patients with patient ID 39 to 72. The first two columns give the gene description and the gene accession number for the particular gene and the successive columns represent the 34 patients. Each patient has two columns - one with the value for each of the 7129 genes and one is the call column which has been explained in the previous section. This total data will be our test set, meaning that the classifier we train will be tested on this data. This will give us a performance measure of our classifier, with parameters such as accuracy, error, loss etc.

4.3 Data Preprocessing:

The dataset does not have any null/empty values and hence no filling was required. The transpose of both the training and test data was done so that the rows represented a patient and the columns the gene expression values. In this way, all gene expressions could be used as features/components in our analysis. But since the number of genes we have is more than seven thousand , it would be a bad idea to train our model treating each gene as a separate feature. Hence we will be applying Principal Component Analysis in the next step of our experiment. Since the data consists of a lot of Biological Science and scientific terms which can be confusing sometimes, also it has nothing to do with training the model, Biological names will be dropped for convenience. Also data enthusiasts reading this can get confused with confusing transcript numbers.So data will be converted into just two columns and the training and further analysis will be done.

4.4 Principal Component Analysis (PCA):

In a dataset, the total number of features determines the dimensionality of it. Having a massive number of features in a dataset often reduces the performance of machine learning algorithms. In Machine Learning, having a large feature space which is n-dimensional and considering rows of data as points in the space usually form a small and non-representative sample. When the model tries to fit over many features, it causes low-performance issues and raises the need for feature reduction.

We will be using Principal Component Analysis (PCA) for emphasizing variation and identifying healthy patterns in our dataset.

What is PCA?

PCA is a dimensionality reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. Reducing the number of variables of a data set naturally comes at the expense of accuracy, but the trick in dimensionality reduction is to trade a little accuracy for simplicity. Because smaller data sets are easier to explore and visualize and make analyzing data much easier and faster for machine learning algorithms without extraneous variables to process. In our case, the number of features/genes being more than seven thousand makes it really unfeasible to train a model taking into consideration all these features.

To deal with this problem, we will use PCA to pack most of our features into tight components which will retain most of the information of our dataset and also discard the features which don't show much correlation. How PCA works is that the most important features are packed in the components at the beginning and the importance of features start decreasing as we move to the second, third and so on components. The new components may not be in a form with any actual meaning/representation but they help in the training and visualisation part of the process. In our case , we will compress our initial 7129 components into 30 Principal Components with the help of PCA and then use KNN to finally build our model and train on these 30 components.

To show a glimpse of what PCA actually does, we applied PCA to our data and reduced it to 3 components.

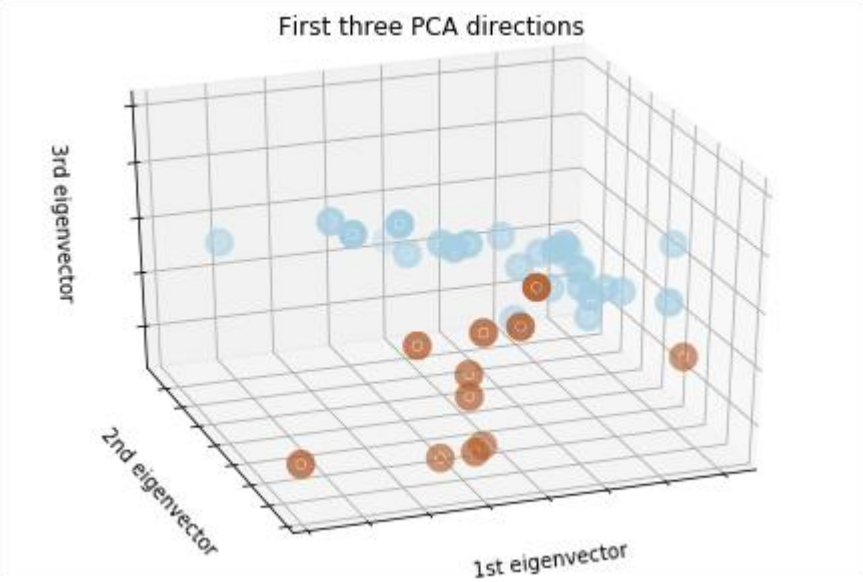

Fig.1. A three dimension breakdown of our dataset with 7129 dimensionality using PCA

As we can see, the two clusters - the brown and the blue clusters representing our two cancer classes (ALL and AML) can be very clearly observed when the dimensionality is reduced using PCA.

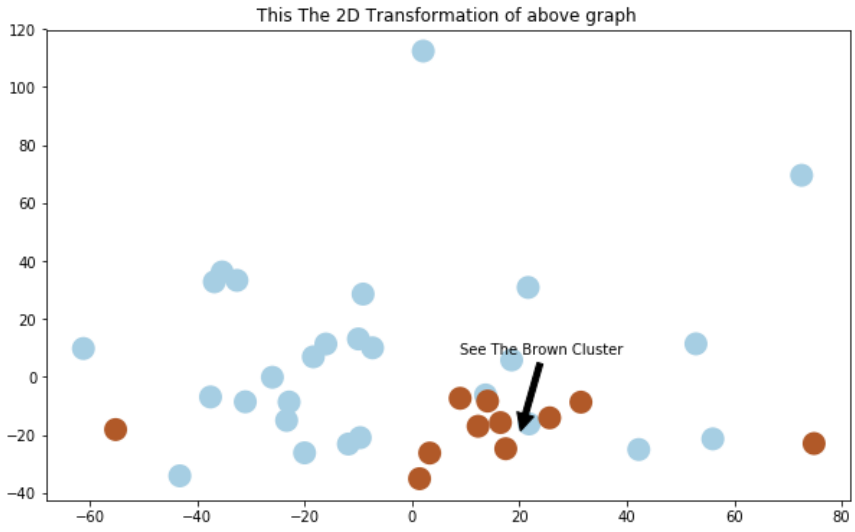

Fig . 2 . 2D representation of the clusters

4.5 Model Building:

After we reduce our dataset to 30 principal components , the data can be visualised like this :-

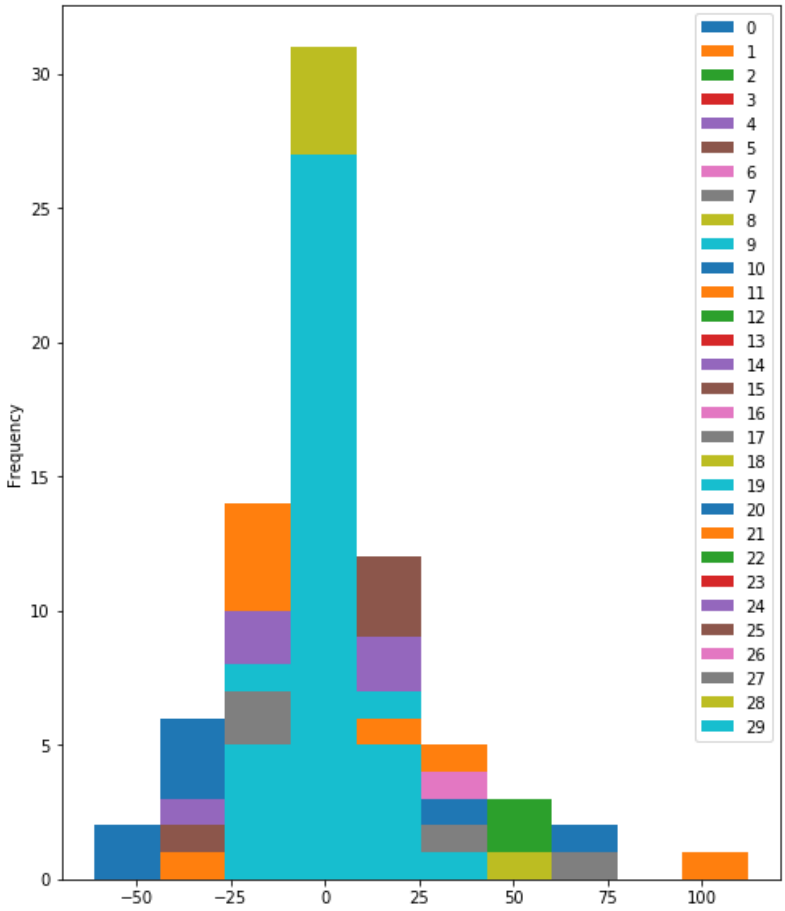

Fig.3. Histogram representing the 30 Principal Components after PCA analysis on data

It is clear from the above histogram that KNN would be a useful approach in our experimentation. The reduction to 3 components did a better job at proving why. If you refer to the images again, you will see that the similar types of cancer tend to cluster together in the three dimensional space. And since reducing 7129 initial components to 3 Principal Components would lead to a lot of data loss, we set the number of final principal components as 30. The reason is, 30 components retain about 95% of the information from our original dataset and these many features can be easily learned by our algorithm. Hence the Principal Component Analysis with the standardisation of our data using a standard scaler completes our data preprocessing portion and we can finally move on to our model training section.

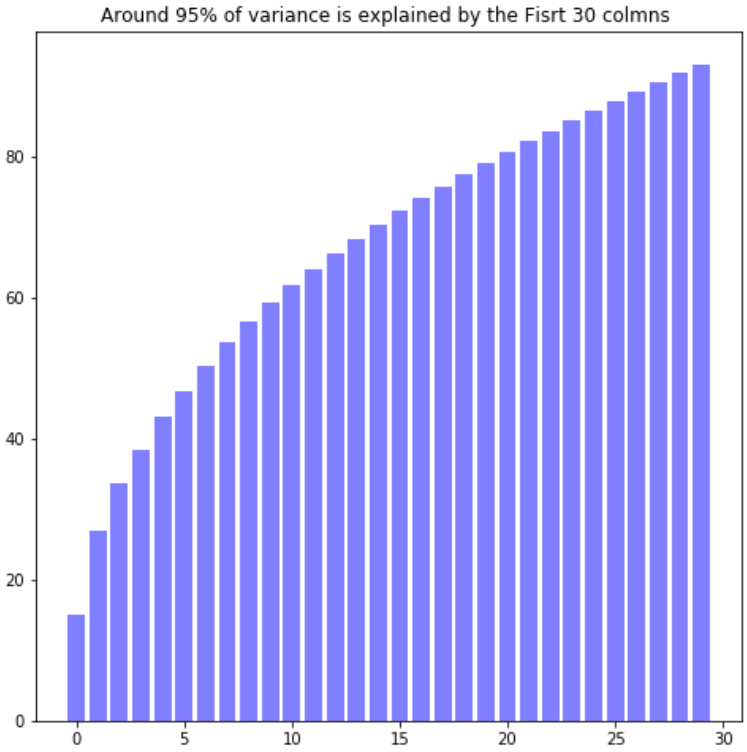

Fig . 4. Only 5% data is lost after reducing 7129 components to 30 components.

4.6 Model Training:

After fitting our 38 training samples to our KNN classifier, we tested the model on the 34 test samples and the following confusion matrix was obtained :

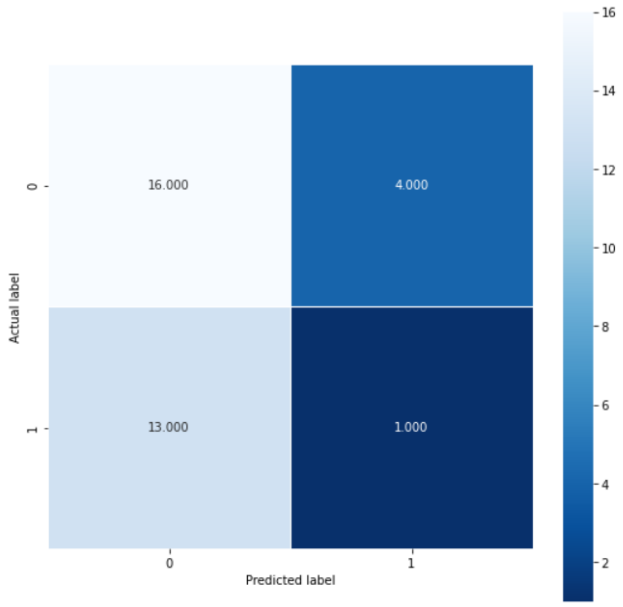

Fig. 5. Confusion Matrix of test results

Let us break down the above matrix to get insights on our model.

5. Model Analysis:

Let us analyse the metrics of our model by breaking down the above confusion matrix. The ALL and AML leukemia classes are represented by the 0 and 1 index on the confusion matrix respectively.

As we can see, out of the 20 ALL samples, 16 were identified correctly by our model which is an 80 percent accuracy when it comes to classifying ALL samples. The rest of the 4 ALL samples were identified incorrectly as AML samples.

When it comes to the AML classification , we see that the results are a little less satisfactory. Out of the 14 AML samples, 13 were classified as ALL samples which means that the model has a lot of trouble differentiating AML from ALL than it has in differentiating ALL from AML.

There could be several reasons for this,

- The training data has more samples corresponding to the ALL class
- May have lead to the model overfitting towards the ALL class
- Poor correlation among principal components
- AML genes not showing any prominent patterns
- Insufficient training data (34 samples only)

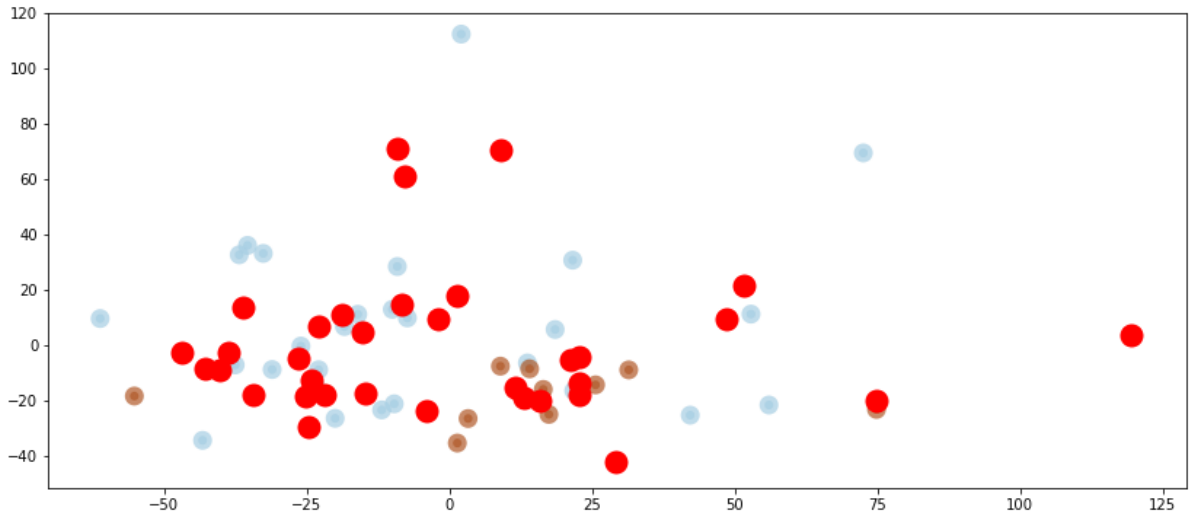

Fig . 6. Plotting the predicted test points vs their actual labels

Here the red points are plotted on the training set to show which points are falsely represented in which category. It can also be thought of as representing the points which were not on the diagonal of the above plotted confusion matrix.

6. Conclusion

The objective of our work was to find a solution to accurately classify the two acute leukemia types - ALL and AML. We started with a literature survey of other works in this field and decided to build a model of our own. The first step was the data collection part of the process. After the data had been collected, we started with pre-processing which is common to all machine learning systems. The distinctive step of our preprocessing was Principal Component Analysis which allowed us to compress a lot of information into relatively less numerous variables and thus eliminate redundant data. After that, we built a machine learning system using KNN and trained our model on that data repetitively. The result was a model which was able to classify the two types of cancer with some level of accuracy. Although we were partially successful in our efforts, the work can be expanded upon if more data is made available at some future time. The system can at one point, even be used by professionals in the medical field to actually classify leukemia instead of doing it manually. That is the power of machine learning that we set out to harness in this work of ours and conclude this chapter with.

7. References

1. Machine Learning Algorithms For Diagnosis Of Leukemia Italia Joseph Maria, T. Devi, D. Ravi INTERNATIONAL JOURNAL OF SCIENTIFIC & TECHNOLOGY RESEARCH VOLUME 9, ISSUE 01, JANUARY 2020 ISSN 2277-8616

2. Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring T. R. Golub1,2,*,†, D. K. Slonim1,†, P. Tamayo1, C. Huard1, M. Gaasenbeek1, J. P. Mesirov1, H. Coller1, M. L. Loh2, J. R. Downing3, M. A. Caligiuri4, C. D. Bloomfield4, E. S. Lander1,5,*

3. Abbas, N., & Mohamad, D. (2014). Automatic color nuclei segmentation of leukocytes for acute leukemia. Research Journal of Applied Sciences, Engineering and Technology, 7(14), 2987–2993.

4. Abbas, N., Mohamad, D., Abdullah, A. H., Saba, T., Al-Rodhaan, M., & Al-Dhelaan, A. (2015). Nuclei segmentation leukocytes in blood smear digital images. Pakistan Journal of Pharmaceutical Sciences, 28(5), 1801–1806.

5. Bibin, D.,Nair, M. S., & Punitha, P.(2017). Malaria parasite detection from peripheral bloodsmearimages usingdeep belief networks.IEEEAccess, 5, 9099–9108

6. Fahad, H. M., Ghani Khan, M. U., Saba, T., Rehman, A., & Iqbal, S. (2018). Microscopic abnormality classification of cardiac murmurs using ANFIS and HMM. Microscopy Research and Technique, 81(5), 449–457. https://doi.org/10.1002/jemt.22998

7. Goutam, D., & Sailaja, S. (2015). Classification of acute myelogenous leukemia in blood microscopic images using supervised classifier. International Journal of Engineering Research & Technology (IJERT), 4(1), 569–574