IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 6, Issue 10, October 2019

Machine Learning as a Decision Aid for Breast Cancer Diagnosis

Rishit Dagli[1]

Student, Thakur International School; Mentor, Young Engineer's Club, Science Kidz Kandivali Mumbai[1]

Abstract: In this paper, we use the diagnosis of breast cytology to demonstrate the applicability of this method to medical diagnosis and decision making. Each of 11 cytological characteristics of breast fine-needle aspirates reported to differ between benign and malignant samples was graded 1 to 10 at the time of sample collection. Nine characteristics were found to differ significantly between benign and malignant samples. Mathematically, these values for each sample were represented by a point in a nine-dimensional space of real variables. We use various different algorithms and also demonstrate the comparison between the algorithms for the classification problem. Finally, an overall accuracy of 99.4048 % is achieved. We only classify 1 % of benign case as malignant. The algorithms used are programmed in python for demonstration purposes. This paper also demonstrates deploying the created model on cloud and building an API for calling the model and verify it.

Keywords: Machine Learning, Decision aid system, Breast Cancer prediction, Logistic Regression, Decision Forest, Neural Network

I. INTRODUCTION

For the following Machine Learning Algorithm, we use the Breast Cancer Wisconsin (Original) Data Set [1]. In this report we present some general Mathematical methods for aiding medical diagnosis and decision-making and demonstrate its application in diagnosing breast mass cytology. We easily infer this as a classification problem as the output variable, here 2 (benign) and 4 (malignant) are discrete values. We use Logistic Regression [2], Multiclass decision forests [3],[4] and Multiclass Artificial Neural Networks [5]. In the present application, the pattern sets were benign or malignant and each element of the pattern sets consisted of nine cytological characteristics of benign or of malignant breast fine-needle aspirates (FNAs). These nine characteristics have been established to differ between benign and malignant samples, but no single described characteristic alone presently distinguishes pattern between benign and malignant samples. Eleven cytological characteristics of breast FNAs were valued on a scale of 1 to 10, with 1 being the closest to benign and 10 the most malignant. Statistical analysis [6] showed that the following nine characteristics differed significantly between benign and malignant samples: uniformity of cell shape, uniformity of cell size, clump thickness, bare nuclei, cell size, normal nucleoli, clump cohesiveness, nuclear chromatin, and mitoses.

II. DATA AND METHODS

A. Logistic regression

We use the Logistic Regression or logit model to model the probability of certain class or event such as pass/fail, win/lose, alive/dead or healthy/sick. This can be extended to model several classes of events. Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although many more complex extensions exist. Analogous models with a different Sigmoid function instead of the logistic function can also be used, such as the Probit model [7]; the defining characteristic of the logistic model is that increasing one of the independent variables multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter; for a binary dependent variable this generalizes the odds ratio. The logistic regression can be understood as finding the best β parameters that best fit:

$$y = \begin{cases} 1, & \beta_0 + \beta_1 + \varepsilon > 0 \\ 0, & \text{else} \end{cases}$$

Where ε is an error distributed by standard logistic distribution.

So, we can define logistic function as

$$\sigma(z) = \frac{e^z}{1 + e^z} = \frac{1}{1 + e^{-z}} \quad \text{where } z \in \mathbb{R}$$

We also infer that $\sigma(z) \in (0,1)$

DOI 10.17148/IARJSET.2019.61002

We can solve this by gradient descent so,

$$\theta_j = \theta_j - \alpha \frac{\partial E}{\partial \theta_j}$$

Here E should be convex or as convex as possible, should be a function of θ and should be differentiable for us to apply gradient descent. So, define

$$\mathrm{CE}_1(\text{cross entropy}) = -p\log(q) = -y\log(\hat{y})$$

Where $\hat{y}$ are the predicted likelihoods so $\hat{y} \in (0,1)$. Also

$$\mathrm{CE}_2 = (1-y)(1-\log(\hat{y}))$$

We simply observe that in a test use case, $\mathrm{CE}_1 + \mathrm{CE}_2$ gives us the expected output. So, define

$$\mathrm{BCE} = \mathrm{CE}_1 + \mathrm{CE}_2 = -y\log(\hat{y}) - (1-y)(1-\log(\hat{y}))$$

Substituting above found values and by chain rule,

$$\frac{\partial \mathrm{BCE}}{\partial \theta_j} = \left(\frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}}\right)\left(\frac{e^{-z}}{(1+e^{-z})(1+e^{-z})}\right)x_j$$

Now we can define the gradient descent as,

$$\theta_j = \theta_j - \alpha\sum_{m=1}^{n}(\hat{y}-y)\ x_j = \theta_j - \frac{1}{m}\sum_{1}^{m} -2(y-\hat{y})$$

Expressing this in a matrix format:

$$\theta_j = \theta_j - \frac{\alpha}{m}(\hat{y}-y)^T \cdot X$$

B. Multiclass decision forest

In a decision forest at each node, feature x_i and threshold a are chosen to minimize resulting 'diversity' in the children nodes. This diversity can be measured by Gini Index. $G(N) = 1 - \sum p(i)^2$

The subdivision continues until every node at the bottom has only one class in it, assigned as a prediction to input x_i [8]. To choose the node or split the decision tree we can define information gain. Let T denote set of training examples $(x, y) = (x_1, x_2 \ldots . x_n, y)$ For a value v taken by an attribute let $S_a(v) = \{x \in T \mid x_a = v\}$

Then the information gain (IG) of T attribute for a is the difference between priori Shannon entropy H(T) and conditional entropy $H(T \mid a)$

$$\mathrm{IG}(T,a) = H(T) - \sum_{v \in \mathrm{vals}(a)} \frac{|S_a(v)|}{T} \cdot H(S_a(v))$$

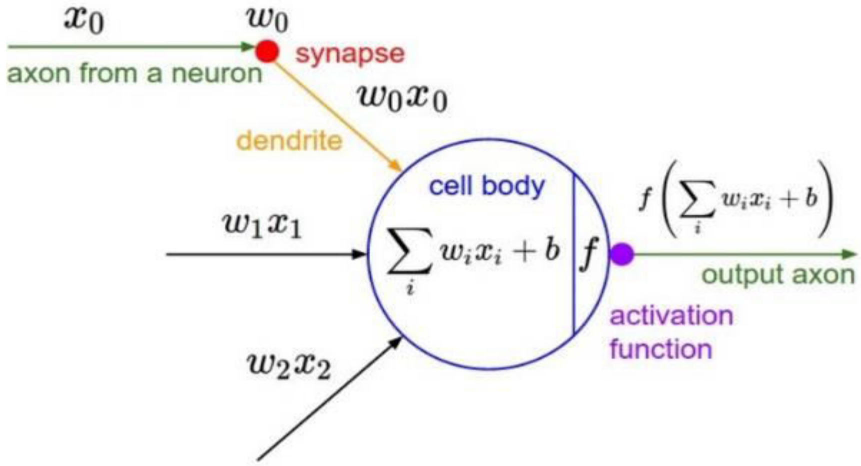
$$\Rightarrow \mathrm{IG}(T,a) = H(T) - E_p|H(S_a(v))|$$

$$\Rightarrow \mathrm{IG}(T,a) = H(T) - H(T \mid a)$$

So, we can choose the split which gives us maximum information gain and set a threshold for a leaf node.

C. Artificial Neural Networks:

An ANN is based on a collection of connected units or nodes called artificial neurons , which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron that receives a signal then processes it and can signal neurons connected to it. In ANN implementations, the "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The figure shows an analogy between the nerve cell and artificial neurons.[9],[10]



As an example, if we have n inputs in a neural network, then with sigmoid activation function the value returned by it will be-

$$y = \frac{1}{1 + e^{-x_{n+1}}}$$

Where $x_{n+1} = \sum_{m=0}^{n} x_m \beta_m$ and β are the weights. The goal here is to optimise these weights. While forward propagation we initialise random weights or β and get the error and while the back propagation we update these weights according to the error [11],[12]. For simplicity we first take 2 layers with the weight in first layer as w_1 and that in the second layer as w_2. We also define z_2 and z_3 as summation of $\beta_n X_n$, let a_n be output from the sigmoid function and for simplicity let $g(x)$ refer to the sigmoid function. So, the error can be calculated as-

$$\text{Error} = \varepsilon = \frac{(y - \hat{y})^2}{2} \cdots\cdots\; (*)$$

For gradient descent we can state -

$$w_{new} = w_{old} - \alpha(\frac{\partial \varepsilon}{\partial w})$$

Now we perform back propagation and find the error ε

$$\frac{\partial \varepsilon}{\partial w_2} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_2}$$

By Chain rule we obtain -

$$\frac{\partial \varepsilon}{\partial w_2} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial z_3} \cdot \frac{\partial z_3}{\partial w_2} = -(y - \hat{y}) g'(z_3) a_2$$

Simplifying this –

$$\frac{\partial \varepsilon}{\partial w_2} = -(y - \hat{y}) g(z_2)\big(1 - g(z_3)\big) \cdot a_2 \;\cdots\cdots\; (**)$$

Let, $\delta_3 = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial z_3}$

Then back propagating till the first layer gives us-

$$\frac{\partial \varepsilon}{\partial w_1} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial w_1} = -(y - \hat{y}) \frac{\partial \hat{y}}{\partial z_3} \frac{\partial z_3}{\partial w_1}$$

$$\Rightarrow \frac{\partial \varepsilon}{\partial w_1} = \delta_3 \cdot \frac{\partial z_3}{\partial a_2} \frac{\partial a_2}{\partial w_1} = \delta_3 \cdot w_2 \cdot \frac{\partial a_2}{\partial w_1}$$

$$\Rightarrow \frac{\partial \varepsilon}{\partial w_1} = \delta_3 \cdot w_2 \cdot g'(z) \cdot X$$

So, we can conclude

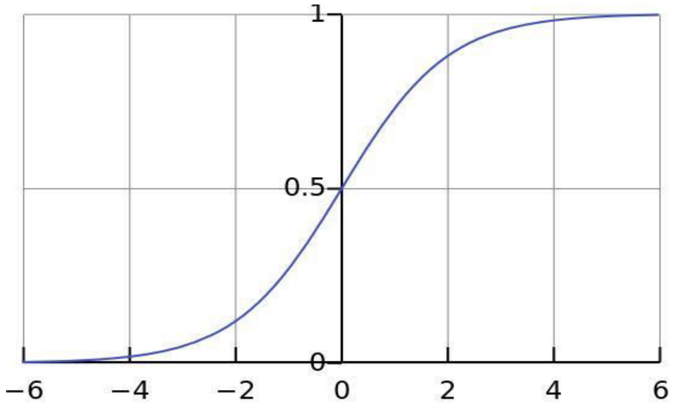
$$\frac{\partial \varepsilon}{\partial w_1} = \delta_3 \cdot w_2 \cdot g(z_2)(1 - g(z_2)) \cdot X \;\cdots\cdots\; (***)$$

By (**) and (***) we can now build the gradient descent formula –

$$w_{2[n]} = w_{2[0]} - \alpha \cdot \frac{\partial \varepsilon}{\partial w_2}$$

$$w_{1[n]} = w_{1[0]} - \alpha \cdot \frac{\partial \varepsilon}{\partial w_1}$$

And so on for each layer in the neural network. A sigmoid activation function works well in this case as it tends to bring the activations to either side of the, making clear distinctions on prediction. Unlike linear function, the output of the activation function is always going to be in range $(0,1)$ compared to $(-\infty, \infty\,)$ of linear function.



DOI 10.17148/IARJSET.2019.61002

Which also makes it an activation function to be used for the case. We do not use ReLu (Rectifier Linear Unit) for the mentioned case as-

$$\text{ReLu} = \begin{cases} 1, & x \geq t \\ 0, & x < t \end{cases}$$

Where t is a threshold value.

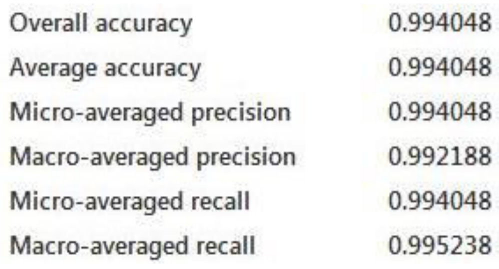
We here face a dying ReLu problem as for activations in a region of ReLu give us the gradient as 0 which means the weights will not be updated and the neurons will stop responding to variations. So ReLu is not used for the mentioned case [13].

III. IMPLEMENTATION

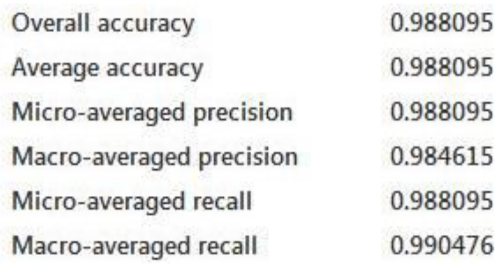
The following algorithms were then programmed in Python. I have also created some comparative visualizations between the above-mentioned suitable algorithms which can aid in choosing the algorithm best for this study [14]. I have also applied Linear Discriminant Analysis techniques on the dataset and found that dimension reduction was not possible. If $P_n = XV_n$ be n^{th} principal component where $[\lambda_1, \lambda_2 \ldots \lambda_n][V_1, V_2 \ldots V_n] = 0$ and $[\lambda_1, \lambda_2 \ldots \lambda_n]$ is a matrix of Eigen values sorted in descending order [15]. We obtained that $P_1, P_2 \ldots P_9$ did not differ by a vast margin. We use the neural network with the hyperparameters configured as following. Single parameter training mode, with 200 hidden layers, $\alpha = 0.001$, set the initial weights to 0.1 and used a min-max normalizer. The decision tree giving the maximum overall accuracy was configured as following – 15 decision trees, set the maximum depth as 32 and 128 random number of splits. Here a train test split is also created to prevent the overfitting of model to the data. I then created a webhook and an API to access the algorithm on edge for both request/response and batch execution and deployed the algorithm on the Azure Cloud.

IV. RESULTS AND DISCUSSION

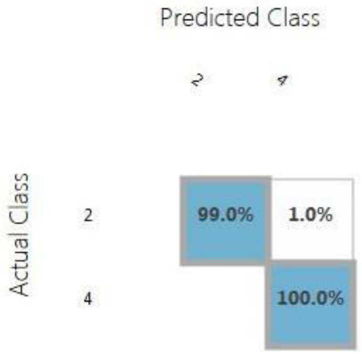
A visual comparison between the Random forest method and Neural Networks obtained, here the left side of the figures represent neural network-based approach and the right side represents random forests-based approach.

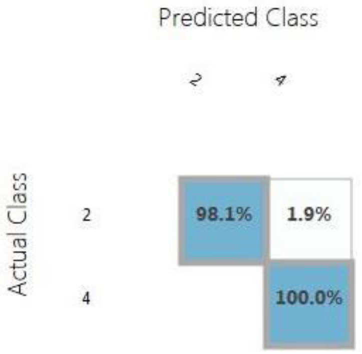


Overall accuracy	0.994048
Average accuracy	0.994048
Micro-averaged precision	0.994048
Macro-averaged precision	0.992188
Micro-averaged recall	0.994048
Macro-averaged recall	0.995238

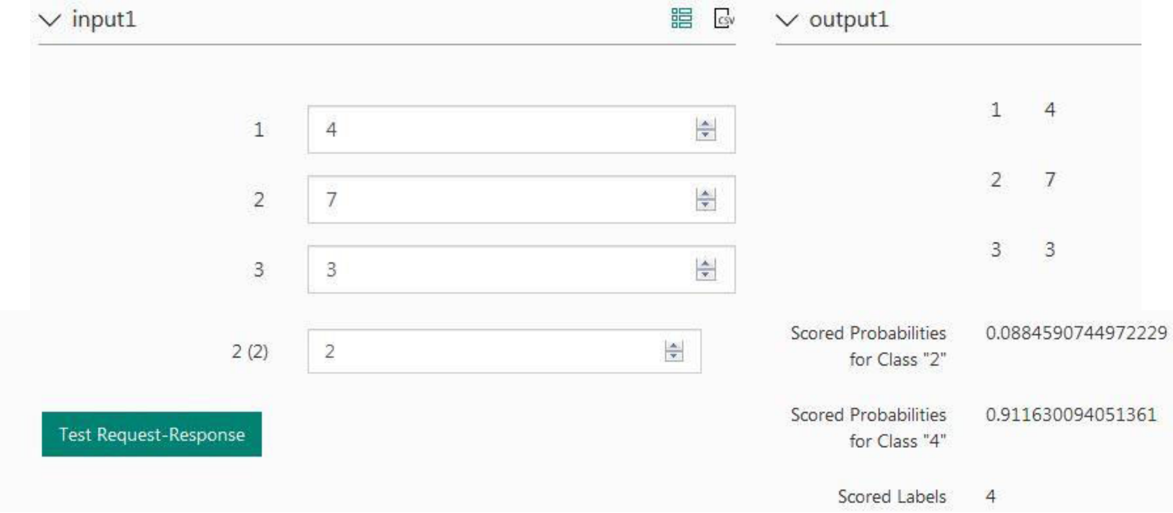


Overall accuracy	0.988095
Average accuracy	0.988095
Micro-averaged precision	0.988095
Macro-averaged precision	0.984615
Micro-averaged recall	0.988095
Macro-averaged recall	0.990476





We observe in the confusion matrix that the false negatives in the random forest-based approach is 1.9% of the test dataset whereas the false negatives are only 1% of the test dataset in the neural network base approach. So, the neural network-based approach becomes the obvious choice and we achieve an overall accuracy of 99.4048 % with only 1% false negative rate. This is the web service deployed for the algorithm (request-response). Here the probabilities of each class label are also displayed.



input1

1: 4

2: 7

3: 3

2 (2): 2

Test Request-Response

output1

1 4

2 7

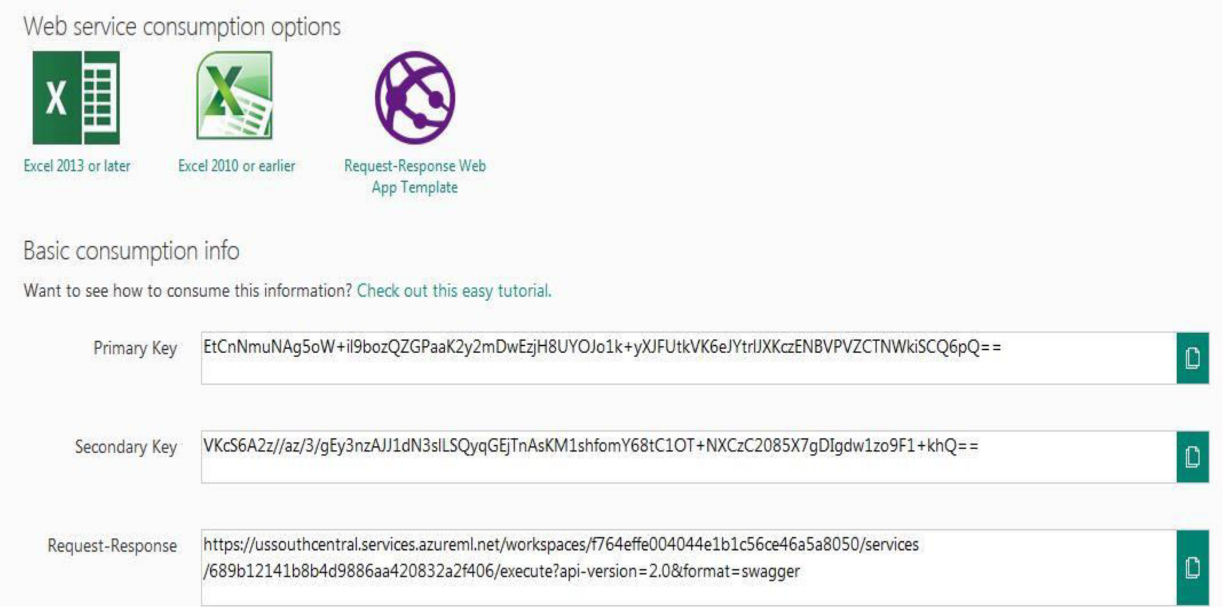
3 3

Scored Probabilities for Class "2": 0.0884590744972229

Scored Probabilities for Class "4": 0.911630094051361

Scored Labels: 4

The model can also be consumed in Excel Sheet or code by using keys [16].



Web service consumption options

Excel 2013 or later

Excel 2010 or earlier

Request-Response Web App Template

Basic consumption info

Want to see how to consume this information? Check out this easy tutorial.

Primary Key: EtCnNmuNAg5oW+il9bozQZGPaaK2y2mDwEzjH8UYOJo1k+yXJFUtkVK6eJYtrlJXKczENBVPVZCTNWkiSCQ6pQ==

Secondary Key: VKcS6A2z//az/3/gEy3nzAJJ1dN3slLSQyqGEjTnAsKM1shfomY68tC1OT+NXCzC2085X7gDIgdw1zo9F1+khQ==

Request-Response: https://ussouthcentral.services.azureml.net/workspaces/f764effe004044e1b1c56ce46a5a8050/services/689b12141b8b4d9886aa420832a2f406/execute?api-version=2.0&format=swagger

REFERENCES

[1]. William H. Wolberg. (n.d.). Retrieved from archive.ics.uci.edu/ml/datasets/breast+cancer+wisconsin+(original).

[2]. Tolles, Juliana; Meurer, William J (2016). "Logistic Regression Relating Patient Characteristics to Outcomes". JAMA JAMA. 316 (5): 533. ISSN 0098-7484. OCLC 6823603312

[3]. Ho, Tin Kam (1995). Random Decision Forests (PDF). Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, 14–16 August 1995. pp. 278–282. Archived from the original (PDF) on 17 April 2016. Retrieved 5 June 2016.

[4]. Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi:10.1109/34.709601

[5]. Bethge, Matthias; Ecker, Alexander S.; Gatys, Leon A. (26 August 2015). "A Neural Algorithm of Artistic Style". arXiv:1508.06576

[6]. Wolberg, W.H. ,Tanner , M.A. & Loh, W.-Y. (1988) Anal. Quant. Cytol. Histol. 10,225-228.

[7]. Liao, Tim Futing (1994). Interpreting Probability Models: Logit, Probit, and Other Generalized Linear Models. Sage. ISBN 0-8039-4999-5.

[8]. Hong, Huixiao & Tong, Weida & Perkins, Roger & Fang, Hong & Xie, Qian & Shi, Leming. (2004). Multiclass Decision Forest—A Novel Pattern Recognition Method for Multiclass Classification in Microarray Data Analysis. DNA and cell biology. 23. 685-94. 10.1089/1044549042476839.

[9]. Aurlien Gron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, O'Reilly Media, Inc., 2017

[10]. Robert R. Trippi , Efraim Turban, Neural Networks in Finance and Investing: Using Artificial Intelligence to Improve Real World Performance, McGraw-Hill, Inc., New York, NY, 1992

[11]. Ian H. Witten , Eibe Frank , Mark A. Hall, Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann Publishers Inc., San Francisco, CA, 2011

[12]. Goodfellow, I., Bengio, Y., & Courville, A. (2017). Deep learning. Cambridge, MA: MIT Press.

[13]. K. Hara, D. Saito and H. Shouno, "Analysis of function of rectified linear unit used in deep learning," 2015 International Joint Conference on Neural Networks (IJCNN), Killarney, 2015, pp. 1-8. doi: 10.1109/IJCNN.2015.7280578

[14]. github.com/Rishit-dagli/Breast-cancer-prediction-ML-Python

[15]. J. Shlens, "A tutorial on principal component analysis", Systems Neurobiology Laboratory Salk Institute for Biological Studies, 2005.
[16]. gallery.azure.ai/Experiment/Breast-cancer-dataset
[17]. I. H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, San Francisco, CA, USA:Morgan Kaufoann Publishers Inc., 2011.
[18]. William H. Wolberg and O.L. Mangasarian: "Multisurface method of pattern separation for medical diagnosis applied to breast cytology", Proceedings of the National Academy of Sciences, U.S.A., Volume 87, December 1990, pp 9193-9196.

BIOGRAPHY

Rishit Dagli is a former TED-X and Ted-Ed speaker. He is a student at Thakur International School and has also written some other research papers specifically in field of Computers, Maths and robotics. He is also a Google Certified mobile site developer, GCP champ, Intel certified AI expert and a cloud enthusiast too. With this he is also passionate about hacking and has also pen tested various websites. Currently he is a mentor at Young Engineer’s Club, a maker’s community.
Website – www.rishitdagli.ml
LinkedIn - www.linkedin.com/in/rishit-dagli-440113165/

DOI 10.17148/IARJSET.2019.61002