IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Server status Monitoring using Advanced Machine Learning Algorithms

[1]Swapneel Bose, [2]Rakesh K.R

[1]Student, [2]Assistant Professor,
[1]Department of Telecommunication Engineering,
[1]RV College of Engineering, Mysore road, Bengaluru-560059, India

Abstract: This paper shows a novel machine learning algorithm known as the K Nearest neighbor (KNN), which is used to classify data into different groups by training the computer to similar data points which exist in close proximity to each other. The machine learning model is used to classify the status of a server as healthy or unhealthy based on physical characteristics such as system memory, swap utility and RAM. Using a prerequisite training data set the machine learning model can perform complex data classification. This algorithm is implemented in Python and the results are visualized in the form of a bar chart using nvd3.js. The entire program is exposed as a web service using flask and corresponding data classification themes and machine learning concepts are explored.

Index Terms **- Machine Learning, K nearest neighbor, python, flask, data visualization, Euclidean distance.**

I. INTRODUCTION

Machine learning is the application of artificial intelligence (AI) which provides computer systems the ability to automatically learn and improve from experience without being explicitly programmed. There are two types of machine learning algorithms, namely, Supervised and Unsupervised.

A supervised machine learning algorithm relies on labelled input data to learn any function that produces an appropriate output when given new unlabeled data. Supervised machine learning algorithms are used to solve classification or regression problems. Classification is the process of predicting the class of given data points based on assumptions made over a prerequisite set of data. Regression returns a single output value based on assumption made on the dataset.

In an Unsupervised machine learning algorithm, the data is unlabeled. i.e., the data is in its raw form. The machine learning model learns on its own and discovers information, data and patterns. In Unsupervised learning there is an input variable (x) but there is no output variable. Hence, the model detects all possible patterns and outcomes to determine the function. Unsupervised machine learning algorithms are used to solve clustering and association problems. Clustering is grouping a collection of objects in such a way that the objects in the same category are more identical to those belonging to other groups. Association concerns the finding of associations between items within the framework of large commercial databases.

In this project, the K-Nearest neighbor algorithm (KNN) is used, which is one of the most essential classification algorithms under Machine Learning. It belongs to the supervised learning domain and will be used to determine the server status of any system based on its physical properties such as CPU usage, ram, and swap utility.

II. MACHINE LEARNING CONCEPTS

2.1 Machine Learning

Machine Learning is a concept that allows the machine to learn from examples and experience without being explicitly programmed. Machine Learning algorithms take a predefined training dataset based on which it devises a mathematical model. This model can make further predictions on new sets of data. Hence, Machine Learning is an evolution of a regular algorithm. It makes the program "smarter" by enabling them to learn automatically from the data provided. A few features of Machine learning are as defined:

1. Efficiency:

The time taken for a basic algorithm to detect patterns in a dataset can take up to days. i.e., the computational time of standard algorithms are high. However, Machine Learning algorithms can compute patterns in a matter of few minutes hence being branded as an efficient algorithm.

2. Accuracy:

Traditionally, data analysis has always contained the trial and error process, an approach that becomes impractical when dealing with massive datasets. By developing efficient and fast algorithms as well as data-driven models for data processing in real-time, machine learning can generate accurate analysis and results.

2.2 Supervised Learning

These algorithms are trained using labelled examples, such as inputs where the desired output is known. These algorithms receive a set of inputs along with their corresponding outputs and uses this data to model a mathematical function to define the relationship between them. Once this function has been defined the algorithm can make predictions on new sets of data and classify them. Basically, input variables (X) and an output variable (Y) are present from which a supervised algorithm learns the mapping function (f) to satisfy the equation Y = f(X).

Supervised learning can solve two kinds of problems: Classification and Regression. Classification is used to group similar labelled data points in different sections in order to classify them. It is a method used to define the laws that describe how to distinguish the various data points. Linear separability is an important concept under classification. It simply means that similar data points can be separated by a single line to define their classes. The line drawn in between to separate classes is called the decision boundary. The area chosen to define the class is called the decision surface. The decision surface shows that if a data point falls within its boundary a certain class will be assigned. Figure 1 shows these concepts.

Regression, unlike the former, is not entirely used for classifying data points. The major difference between classification and regression is that regression returns a numeric value instead of a class. Therefore, regression is useful when predicting numeric based problems like the temperature for any given day, stock market prices or the probability of an event.

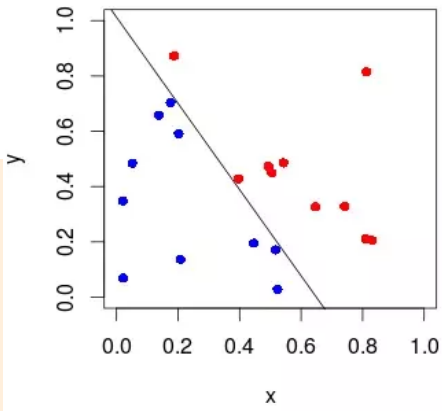

Figure 1. Classification by Linear Separability

2.3 Unsupervised Learning

These algorithms are trained using unlabeled examples, such as inputs where the desired output along with the function are unknown. The learning algorithm receives a set of inputs without their corresponding outputs and uses this data to model a mathematical function to define the relationship between the inputs and outputs. The goal is to explore the data and find some structure within which can define a model. Once this function has been defined the algorithm can make predictions on new sets of data. Basically, just the input variables (X) are provided and a supervised algorithm learns the mapping function from the input to determine function f and output (Y) from Y=f(X).

Unsupervised Learning can solve two types of problems: Clustering and Association. Clustering is the process of creating groups each having a different set of characteristics. Clustering attempts to find different subgroups within a dataset. Since this is unsupervised learning, there is no limit to the number of labels and we are free to choose how many clusters to create. Association is used to determine the rules that describe the data. For example, if a person watches video A, they 're likely to watch video B. Association laws are great for situations where we want to locate similar objects or find relations between two objects.

2.4 K-Nearest Neighbor Algorithm

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. KNN can be used for both classification and regression predictive problems. In this project it is used as a classifier. KNN is a classifier which can take any new set of data (test dataset) and assign it to a group based on assumptions made by a prerequisite set of data (training dataset). The KNN algorithm assumes that similar data points exist in close proximity to each other. To determine whether the data points are in close range to each other or not the KNN algorithm calculates the Euclidean distance between the test data and the training data as shown in Equation 1.

$$D = (\sqrt{x1^2 - y1^2} + \sqrt{x2^2 - y2^2}) \quad \text{(Eq.1)}$$

Where, (x1, x2) is the test set data while (y1, y2) is the training set data

After calculation Euclidean distances the next and most important step in this algorithm, is to select a K-number. The K-number defines the decision boundary which defines the class of a set of data. It is observed that the boundary becomes smoother with increasing value of K. Figure 2 depicts how data obtained from KNN algorithm is represented on free space after being grouped. In the example, K=5 is the chosen K value and the black dot represents a new data point which is classified based on its proximity to the nearest class.

The advantages of using this algorithm are:

1. The algorithm is simple and easy to implement.
2. There is no need to tune several parameters or make additional assumptions.
3. It's a flexible algorithm. It can be used for classification, regression and searching.

However, there are drawbacks to this algorithm. The major and most troublesome drawback is that this algorithm becomes significantly slower as the number of datasets increases. Larger datasets result in longer computational time.

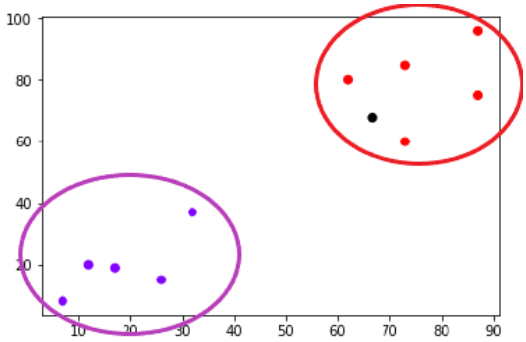

Figure 2. Distribution of data points in KNN

2.5 Flask- REST API

Flask is a micro web framework which provides you with tools, libraries, and technologies that allow you to build a web application. REST stands for REpresentational State Transfer and is an architectural style used in modern web design. It describes the set or rules / constraints for a web application to send and receive data. REST is a set of rules developers follow when they create APIs. API stands for Application Programming Interface, and it refers to the mode of communication between any two software applications. An API is just a medium that lets two entities of code communicate with each other. Hence, REST API's act as a medium to exchange data between a client and a resource. A client can refer to either a developer or software application which uses the API whereas a resource describes an object, data, or piece of information that may need to be stored or sent to other services.

Humans communicate with one another using a common language. Communication would be uninterpretable without a medium of communication. Similarly, APIs have a set of rules for machines to communicate with each other. These rules are called Protocols. HTTP is one such protocol. It is the basis of any data transfer on the Web and a client-server protocol. RESTful APIs almost always rely on HTTP. While working with RESTful APIs, a client will send an HTTP request, and the server will respond with the HTTP response.

JSON (JavaScript Object Notation) format- JSON is a popular format for sending and receiving data over the web. It is based on a subset of the JavaScript programming language and it is easy to understand and generate. It is like a dictionary data type and contains a key along with a token value. The data being exchanged over the web can only be in the form of text. With the help of the Flask library, the system can convert any JavaScript object into JSON, and send JSON to the server.

2.6 Data Visualization – nvd3.js

NVD3.js is a JavaScript visualization library that is free to use and open source. It can be used to represent data in the form of reusable pictorial charts. This library can be extremely powerful for everyday tasks, data representation and even in the corporate sector. NVD3.js is extensively used for real time data visualization of data. The previous section looked into JSON data format. This JSON data is taken as an input in the NVD3.js library and its corresponding data is represented.

In this project, a functional and accurate KNN module is designed and is exposed as a web service using flask. Concurrently, the output of flask is fed into the NVD3.js module in real time and the corresponding JSON data is visualized in the form of a multi-bar chart. This data upon visualization is further exposed as a web service using Flask.

III. METHODOLOGY

3.1 K-Nearest Neighbor

The block diagram of the methodology is given by Figure 3. Two datasets, Test data and Train data are input into the KNN module. The train data is essential as it contains a dataset along with the classification it belongs to. The KNN module studies this dataset to determine the classification of any new set of data. Test data contains the new set of data whose classification is yet to be determined. Hence, with the test and train data input to the KNN module, it can classify new sets of data by implementing its algorithm. After the KNN process is successful in data classification, we use Flask REST API which is a micro framework for creating web applications.

Flask REST API is essential in exposing the KNN algorithm as a web service. Here, the data is expressed in a webpage and it's corresponding JSON data is obtained. This JSON data is then input to the NVD3 data visualization toolkit and the outcome, i.e., a real time graphical representation of the KNN classifier is displayed over a webpage.

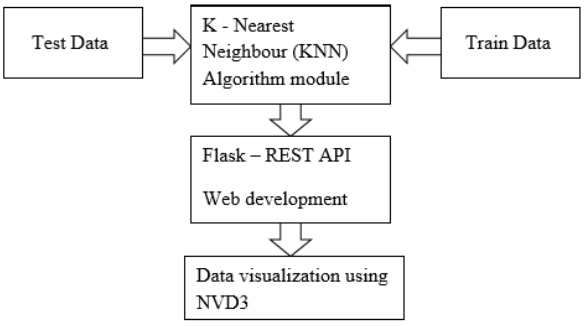

Figure 3. Block diagram of methodology

The algorithm goes through many steps while it is being implemented from scratch. The test and train datasets are primary inputs and serve as a backbone for the algorithm. Certain steps need to be followed in a systematic order while implementing the KNN algorithm. The steps involved in KNN algorithm are:

1. Load the data (Initial dataset is split into test and train set)
2. Calculate the Euclidian distance between the test and train dataset
3. Add the distance and the index of the example to an ordered
5. Select a K value and pick the first K number of entries from the sorted collection
6. Get the labels of the selected K entries
7. Calculate percentage of each classification
8. Return the mode of the K labels

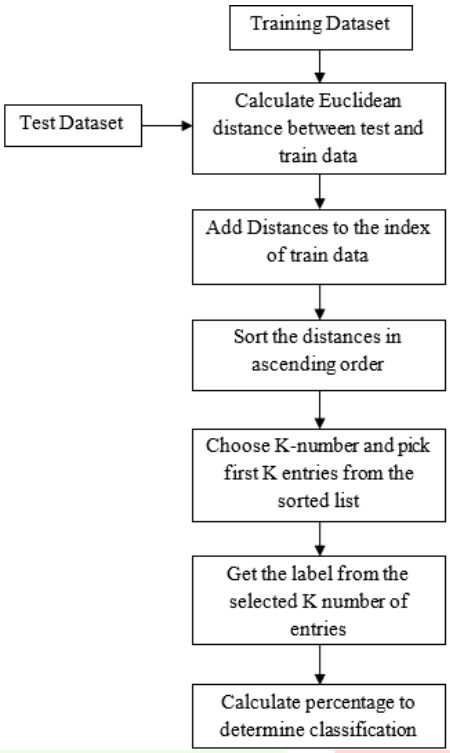

Figure 4. KNN algorithm flowchart

3.2 Flask – ReST API

JSON (JavaScript Object Notation) format- JSON is an extremely popular format for sending and receiving data over the web. It is based on a subset of the JavaScript programming language and it is easy to understand and generate. It is like a dictionary data type and contains a key along with a token value. The data being exchanged over the web can only be in the form of text. With the help of the Flask library, we can convert any JavaScript object into JSON, and send JSON to the server.

3.3 Data Visualization – Nvd3.js

NVD3.js is a JavaScript visualization library that is free to use and open source. It can be used to represent data in the form of reusable pictorial charts. This library can be extremely powerful for everyday tasks, data representation and even in the corporate sector. NVD3.js is extensively used for real time data visualization of data. The previous section looked into JSON data format. This JSON data is taken as an input in the NVD3.js library and its corresponding data is represented.

In this project, a functional and accurate KNN module is designed and is exposed as a web service using flask. Concurrently, the output of flask is fed into the NVD3.js module in real time and the corresponding JSON data is visualized in the form of a multi-bar chart. This data upon visualization is further exposed as a web service using Flask.

IV. IMPLEMENTATION

4.1 Libraries required to implement KNN in Python

The KNN algorithm is simulated on Jupyter notebook (version 6.0.3) which is an open source web application to simulate Python 3 codes. Certain libraries need to be loaded to successfully implement the algorithm. The three most important libraries being used are:

4.1.1 Pandas library

Pandas is a fast, powerful, flexible and easy to use library for the Python programming language, used for open source data analysis and manipulation. Pandas allow importing data of various file formats such as csv, excel etc. The test and train datasets mentioned earlier are of CSV file format. Hence, this is the most important library used as its primary function is to manipulate data in CSV format. The most notable function present within the Pandas library is the ability to take data (like a CSV or Excel, or a SQL database) and create a Python object with rows and columns called data frame that looks very similar to table in a statistical software.

Our program output will be in the form of a data frame. To implement the KNN algorithm the Pandas library will be used to perform the following tasks:
•Opening a CSV file and converting it into a data frame.
•Viewing and Inspecting Data present in the data frame and then manipulating it.
•Selecting specific data and re-arranging by filtering, sorting or grouping.
•Joining/combining two or more data frames together.
•Data cleaning. i.e., removing null values and filling blank spaces to maintain dimensionality of a data frame.

4.1.2 Math Library

It is a library used in Python to implement the basic mathematical functions in a program. In this program the Square root function (sqrt) is extracted from the Math library for the sole purpose of calculating the Euclidean distance.

4.2 Flask Implementation

To implement Flask in Python the results of the KNN program are imported and loaded onto a new python program where the flask infrastructure will be designed. To implement flask and gain access to its various components two libraries are loaded:

4.2.1 JSON Library

The JSON library is primarily used to translate the python dictionary to a JSON string that can be written to a script. While the JSON module converts strings to Python datatypes, the JSON functions are usually used to write JSON data into an external file. JSON can be implemented in many languages, making it suitable for communication between applications. JSON is most widely used for communicating between the web server and client. In this program, The JSON library is used to parse a dictionary to JSON format which will later be used for data visualization.

4.2.2 Flask Library

This library contains all the necessary commands needed to expose a python program as a web service. Two important commands are imported from this library:
•Jsonify() - The arguments to this function are the same as to the dictionary constructor. In other words, this function is used to return a JavaScript Object Notation (JSON) response in Flask which in turn standardizes the format of the response data to be displayed on a webpage.
•Render_template() – It is used to display the HTML template which is stored in the templates folder of the project. Optionally, it can pass variables like the message variable that will be available to the template. In this project, the sole purpose of this command is display the HTML file containing real time data visualization of the project over an active running web server defined by flask.

4.3 Data Visualization Implementation

The KNN algorithm and flask framework were implemented using Python programming language. However, data visualization is executed using Notepad++ (version 7.8.6) which is a text editor for writing JavaScript along with HTML code. To design a reusable multi bar chart to visualize the data four primary steps are involved:

1. Loading the NVD3 library using CDN (Content Delivery Network)
Rather than locally installing the required libraries onto our system these libraries are available in the form of a CDN (Content delivery network) where they can be loaded directly by specifying the URL. CDN's are widely used in HTML and JavaScript applications and makes it easier for the programmer due to fact that there will be no need to use a pip function to install the desired libraries onto the system, and rather just load them from an external URL source. The libraries which are loaded via CDN contain modules to manipulate the graph properties and add additional features. A snippet of CDN's which are added to the head section of the HTML page is shown in Figure 3.2. Here, the libraries nvd3.js, it's native library d3.js along with their corresponding CSS files are loaded.

2. Creating a basic webpage template
The next step is to create a HTML template which specifies the properties of the web page over which our graph will be displayed. Basic webpage properties such as title, background colour, font type, font colour, transitions etc., are defined in this template. Changes made here will be displayed on the webpage.

3. Creating a JavaScript template
Here, a JavaScript file is created which specifies the properties of the graph. This file specifies the function where the data in JSON format is called as an input and its corresponding graph is displayed. Basic graph properties such as graph type (Pie chart, bar graph, line graph, Scatter plot, etc.) are defined here along with other properties such as graph heading, graph colour, axis heading, axis spacing, etc.

4. Creating a CSS file
The CSS file defines the pixel height of the nvd3.js chart which has been designed. It is advisable to increase the pixel height when there is a greater distance between consecutive data values which are being plotted. It is the final file required to create a successfully NVD3.js chart.

These four segments need to be defined by the user in order to visualize the data and display it over a HTML webpage. In this project, rather than coding them in four different files and then importing them over to the main HTML template, Notepad++ was used to create one single block where all the templates along with the CSS file are written. This is done for ease of explanation and to simplify debugging of the program. Once the multi bar chart is fully functional and is displaying accurate results, the render_template() function is used to display this chart on a webpage.

V. SIMULATION RESULTS

This section contains the simulation results. The results of the KNN algorithm to classify new data followed by exposing the program in a flask architecture along with its data visualization are explored here. The original test and train data sets provided by the company have 800+ entries and hence only 10 entries have been extracted from the training dataset which will be displayed.

5.1 Plotting of training dataset

The train dataset which has been supplied by our company contains real time analytical data which determines the classification of a server based on three physical properties namely, CPU usage, swap utility and memory usage.

The KNN algorithm states that similar data points will exist in close proximity to one another. This has been represented in the form of a scatter plot using Microsoft Excel. Figure 4.1 shows a scatter plot which plots the points of all three physical properties specified in the training dataset. It shows how the similar data points are grouped in close proximity to one another and how a decision line is placed to separate both classes.

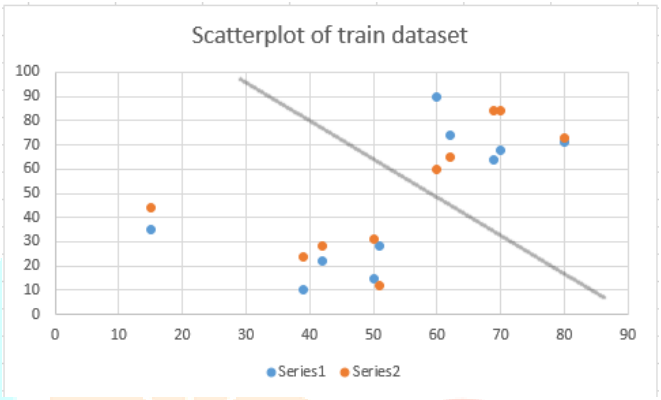

Figure 5. Scatterplot of training dataset

With reference to Figure 5, If any new data point is plotted on this graph and it falls on the boundary above the decision line then it will come under the category “Healthy” and if it falls under the decision line then it will be categorized as “Unhealthy”. This is the principle concept on which the KNN algorithm classifies new data points.

5.2 KNN Algorithm simulation results

The results of the KNN classification module designed is executed on Python. Four results from the Test dataset have been displayed along their respective classifications after being passed through the KNN algorithm.

Figure 6 shows the results for a new dataset with values (65,77,33). The percentage of good is greater and hence the classification is “Good”. Therefore, for a system with CPU usage=65, Swap utility= 77 and Memory usage=33, The overall system performance will be healthy.

	cpu_usage	memory_usage	swap_util	Distances	classification	% good	% bad	New Classification
0	65	77	33	63.820060	0.0	0.6	0.4	good
1	29	66	96	85.246701	0.0			
3	40	73	20	59.874870	0.0			
5	85	36	62	51.254268	1.0			
8	29	30	44	28.896367	1.0			

Figure 6. KNN program result for (65,77,33)

Figure 7 shows the results for a new dataset with values (29,66,96). The percentage of bad is greater and hence the classification is “Bad”. Therefore, for a system with CPU usage=29, Swap utility= 66 and Memory usage=96, The overall system performance will be unhealthy.

	cpu_usage	memory_usage	swap_util	Distances	classification	% good	% bad	New Classification
0	65	77	33	55.118055	0.0	0.4	0.6	bad
3	40	73	20	47.010637	0.0			
5	85	36	62	60.991803	1.0			
6	57	96	58	82.316463	1.0			
8	29	30	44	38.884444	1.0			

Figure 7. KNN program result for (29,66,96)

Figure 8 shows the results for a new dataset with values (40,73,20). The percentage of good is greater and hence the classification is “Good”. Therefore, for a system with CPU usage=40, Swap utility= 73 and Memory usage=20, The overall system performance will be healthy.

	cpu_usage	memory_usage	swap_util	Distances	classification	% good	% bad	New Classification
0	65	77	33	66.219333	0.0	0.8	0.2	good
1	29	66	96	62.136946	0.0			
3	40	73	20	51.429563	0.0			
4	31	73	97	67.149088	0.0			
8	29	30	44	14.866069	1.0			

Figure 8. KNN program result for (40,73,20)

Figure 9 shows the results for a new dataset with values (85,36,62). The percentage of bad is greater and hence the classification is “Bad”. Therefore, for a system with CPU usage=85, Swap utility= 36 and Memory usage=62, The overall system performance will be unhealthy.

	cpu_usage	memory_usage	swap_util	Distances	classification	% good	% bad	New Classification
0						0.2	0.8	bad
4	31	73	97	41.4126	0			
5	85	36	62	41.6293	1			
6	57	96	58	40.3609	1			
7	75	68	96	13	1			
9	47	79	89	25.9808	1			

Figure 9. KNN program result for (85,36,62)

5.3 Flask Simulation Results

The results of the KNN algorithm are exposed as a flask web service and their corresponding results are converted into JSON file format to be used for data visualization. The JSON output for the previously obtained KNN program is displayed in Figure 10. The data shown in Figure 4.6 is used as an input for Data visualization.

[{"good": 60, "bad": 40}, {"good": 40, "bad": 60}, {"good": 60, "bad": 40}, {"good": 80, "bad": 20},
{"good": 40, "bad": 60}, {"good": 40, "bad": 60}]

{"good": 40, "bad": 60}, {"good": 20, "bad": 80}, {"good": 40, "bad": 60}, {"good": 20, "bad": 80},

Figure 10. KNN Output in JSON format

5.4 Data Visualization

Data visualization is performed using nvd3.js. Here, the percentages of each training data is displayed over a multi bar chart. From this data, we can observe the variation is percentage values. The higher percentage determines the class of the new dataset. The 10 values which have been extracted are displayed in Figure 11. It is observed that each value is verified from the KNN program results and the higher percentage of each data point determines its class; Healthy or Unhealthy.

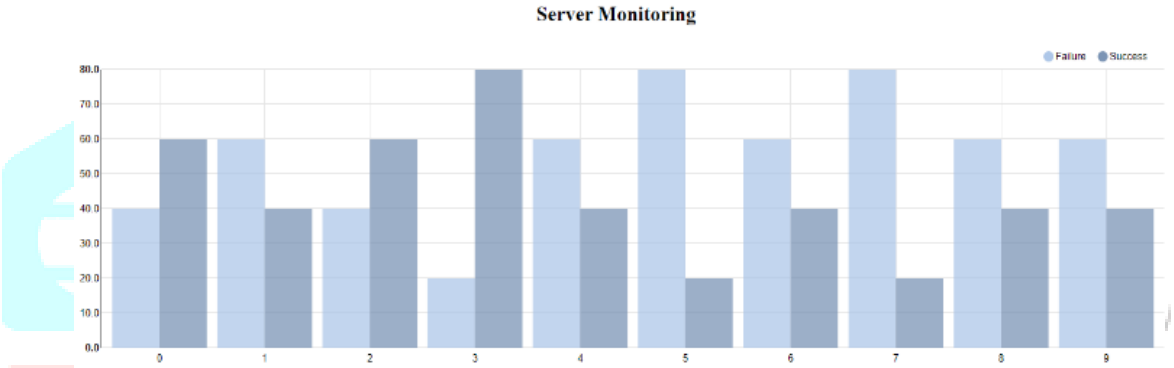

Figure 11. Data visualization results

VI. CONCLUSION AND FUTURE SCOPE

Machine Learning algorithms are improving with an increase in demand and their multiple applications. KNN is an effective machine learning algorithm that can be used in credit scoring, prediction of cancer cells, image recognition, and many other applications which involve pattern recognition. The main importance of using KNN is that it's easy to implement and works well with small datasets with accurate results. It doesn't work well with large datasets due to longer computation timing and increased memory consumption.

The scope for future enhancement introduces research introduces modifications in the KNN algorithm to mitigate the drawbacks of increased computation time and memory. One such proposed method is to blend different algorithms or techniques that combine the strengths of individual algorithms. Efforts should be taken to combine two or more of these machine learning algorithms to overcome their respective drawbacks. Methods that exploit the strengths of techniques such as neural networks and SVM can be explored.

REFERENCES

[1] Mohammadi F.G., Amini M.H, and Arabnia H.R. "An Introduction to Advanced Machine Learning: Meta Learning Algorithms, Applications, and Promises.", Optimization, Learning, and Control for Interdependent Complex Networks. Advances in Intelligent Systems and Computing, vol. 1123, 2020.

[2] Binkhonain, M., & Zhao, L. "A review of machine learning algorithms for identification and classification of non-functional requirements. Expert Systems with Applications", 2019.

[3] Kunal Renan. "Building REST APIs with Flask: Create Python Web Services with MySQL", 2019

[4] Gupta, V.K. "An Analysis of Data Visualization Tools. International Journal of Computer Applications" 178, pp. 4-7, 2019

[5] Chauhan, N., Singh, M., Verma, A., Parasher, A., & Budhiraja, G. "Implementation of database using python flask framework: college database management system". International Journal of Engineering and Computer Science (2019)

[6] Vangala Rama Vyshnavi et al. International Journal of Recent Research Aspects ISSN: 2349-7688, Vol. 6, Issue 2, pp. 16-19, June 2019.

[7] Gou, J., Ma, H., Ou, W., Zeng, S., Rao, Y., & Yang, H. "A generalized mean distance-based k-nearest neighbor classifier". Expert Systems with Applications, 2018

[8] Kurilovas, E. "Advanced machine learning approaches to personalize learning: learning analytics and decision making, Behavior & Information Technology" pp. 1–12, 2018.

[9] Chakraborty, D., & Elzarka, H. "Advanced machine learning techniques for building performance simulation: a comparative analysis". Journal of Building Performance Simulation pp. 1–15, 2018.

[10] Triguero, I., García-Gil, D., Maillo, J., Luengo, J., García, S., & Herrera, F. "Transforming big data into smart data: An insight on the use of the k-nearest neighbors algorithm to obtain quality data". Wiley Interdisciplinary Reviews: Data Mining and Knowledge, 2018

[11] Nitin Kumar Karma. "RESTFul Web Services - The Python Flask Way: Build RESTful APIs using Python and Flask-Restful". Independently published (2018)

[12] J. Liu, T. Tang, W. Wang, B. Xu, X. Kong and F. Xia, "A Survey of Scholarly Data Visualization," in IEEE Access, vol. 6, pp. 19205-19221, 2018

[13] Yang, K., Cai, Y., Cai, Z., Xie, H., Wong, T.-L., & Chan, W. H. "Top K representative: a method to select representative samples based on K nearest neighbors". International Journal of Machine Learning and Cybernetics, 2017

[14] S.M Sohan., Frank Maurer., Craig Anslow., Martin P. Robillard."A Study of the Effectiveness of Usage Examples in REST API Documentation". IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), 2017

[15] D. Keim, H. Qu and K. Ma. "Big-Data Visualization," in IEEE Computer Graphics and Applications. vol. 33, no. 4, pp. 20-21, 2013