SPL-1 Project Report

Machine Learning Algorithms

Submitted by

Abir Ashab Niloy

BSSE Roll No. **: 1315**

BSSE Session: **20-21**

Submitted to

Kishan Kumar Ganguly

Assistant Professor, IIT

University of Dhaka

Institute of Information Technology

University of Dhaka

[21-05-2023]

Project : Implementation of Machine Learning Algorithms in c/c++

Author : Abir Ashab Niloy

-BSSE 13th Batch

-(Roll - 1315, Exam Roll - 115525)

Date Submitted : May 21th, 2023

Supervised By : Kishan Kumar Ganguly

Assistant Professor

Institute of Information Technology,

University of Dhaka.

Supervisor's Approval :

____

(Signature of Kishar Kumar Ganguly)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to my project supervisor, Kishan Kumar Ganguly Sir, Assistant Professor, Institute of Information Technology, University of Dhaka, for giving me an opportunity to go on with this project and providing unconditional support throughout the semester. His invaluable guidance and motivation helped me make this project a successful one. It was an honor to work under his supervision.

Project Availability : https://github.com/Abir-Ashab

Table of Contents

1. Introduction

Welcome to my project on Machine Learning Algorithms in C/C++! In this endeavor, I delve into the fascinating world of machine learning and explore its applications through the lens of C/C++ programming. Through this project, I aim to demonstrate the power and versatility of machine learning algorithms implemented in these languages.

Our focus lies in providing a concise yet comprehensive overview of various popular machine learning algorithms, including but not limited to decision trees, random forest,Kmeans, and k-nearest neighbors. I will showcase the implementation of these algorithms in C/C++, highlighting their efficiency and effectiveness.

By harnessing the combined strength of C/C++ and machine learning, we unlock endless possibilities for developers and researchers to leverage these algorithms in real-world scenarios. Whether you are a beginner eager to explore the fundamentals or a seasoned programmer seeking to expand your repertoire, this project will serve as an invaluable resource.

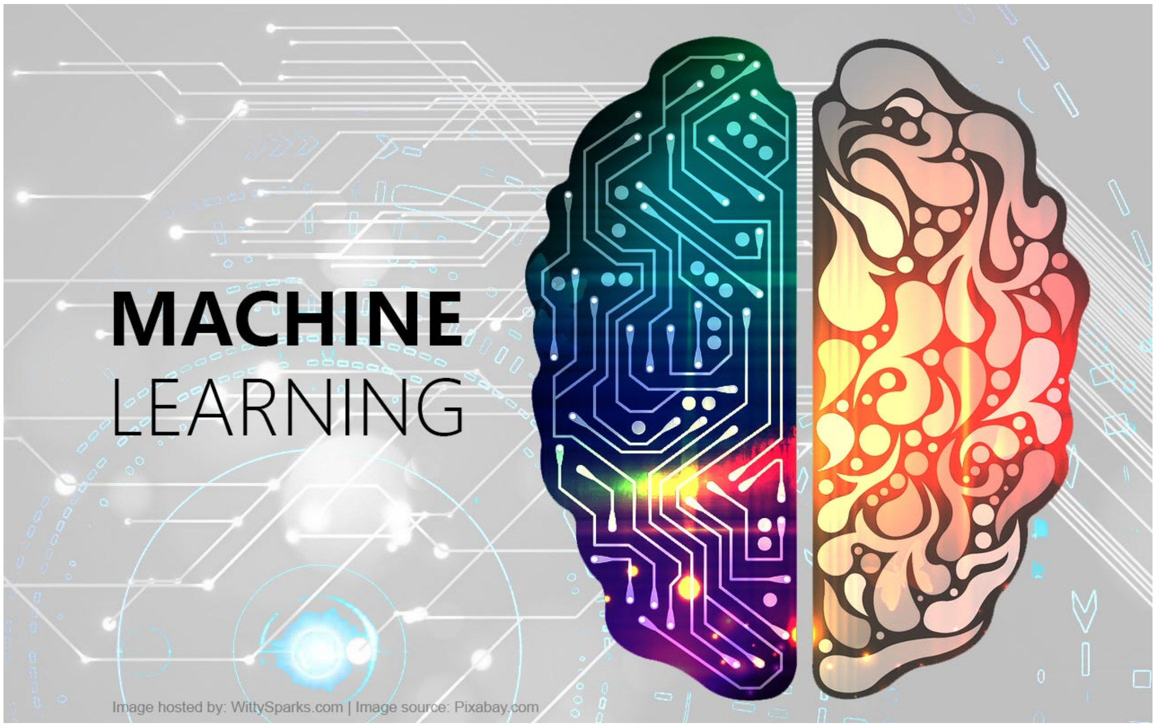
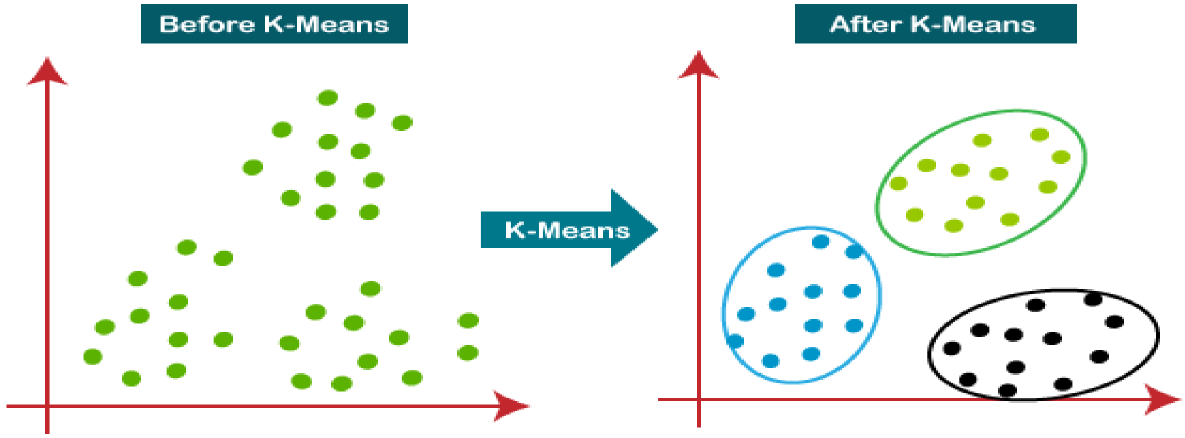


Image hosted by: WittySparks.com | Image source: Pixabay.com

2. Background Study : Here i will discuss regarding my journey of study to implement this wonderful project;

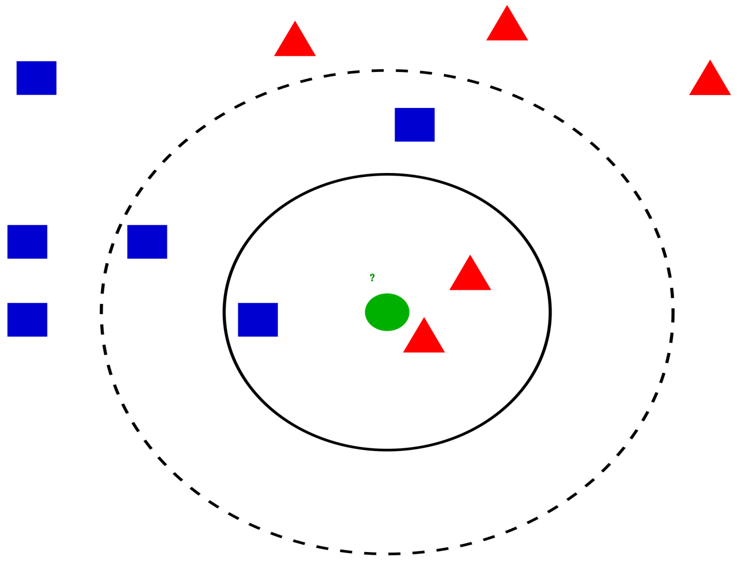
2.1 K Means

The background study to implement the K-means algorithm in C/C++ involves understanding the fundamentals of the K-means clustering algorithm, its mathematical foundations, and the principles behind its implementation. This includes comprehending concepts like Euclidean distance, centroid initialization, iterative refinement, and convergence criteria. Additionally, gaining familiarity with C/C++ programming concepts such as data structures, loops, and arrays is crucial for effectively implementing the algorithm.



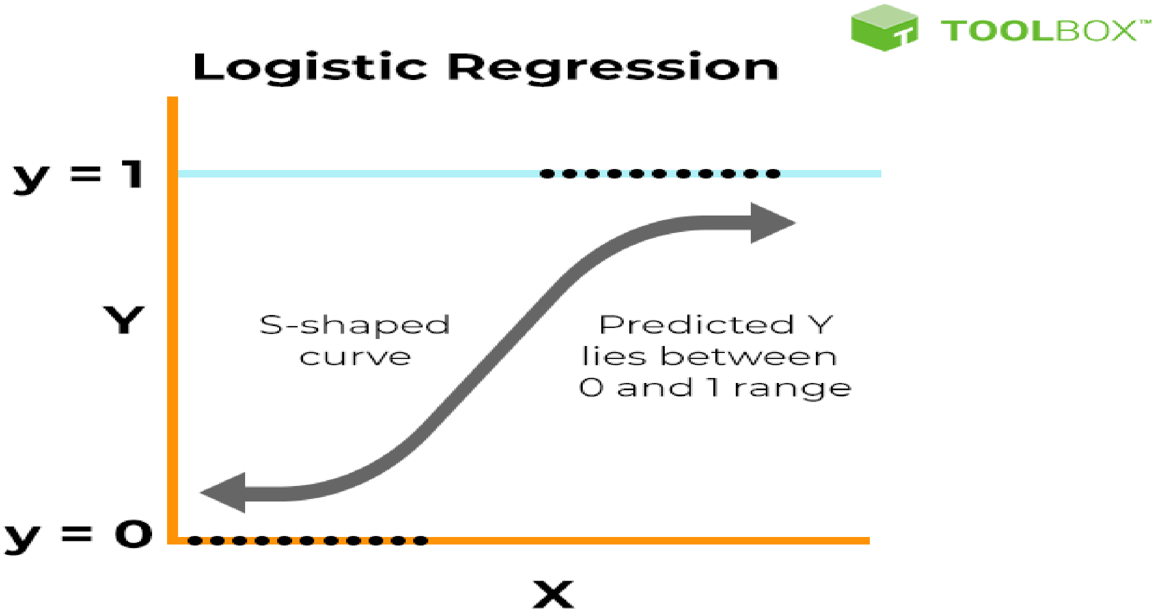
2.2 KNN

To implement the K-Nearest Neighbors (KNN) algorithm in C/C++, a background study is crucial. This involves understanding the fundamental principles of KNN, such as distance metrics, classification, and the concept of K nearest neighbors. Additionally, exploring C/C++ libraries for handling arrays and data structures, as well as familiarizing oneself with relevant programming techniques for efficient computation, will ensure a successful implementation of the KNN algorithm in C/C++.



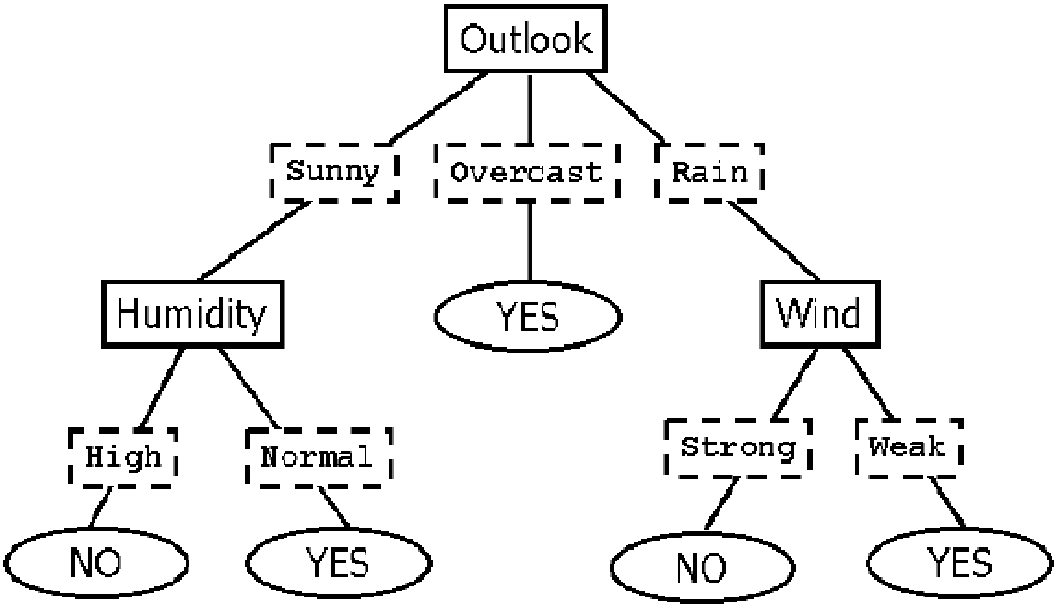
2.3 Logistic Regression

The background study to implement the logistic regression algorithm in C/C++ involves understanding the mathematical concepts behind logistic regression, specifically the logistic function and maximum likelihood estimation. It also requires familiarity with programming concepts in C/C++ such as data structures, handling numerical computations, and optimization techniques. Additionally, studying existing implementations and libraries in C/C++ can provide insights and guidance for the implementation process.



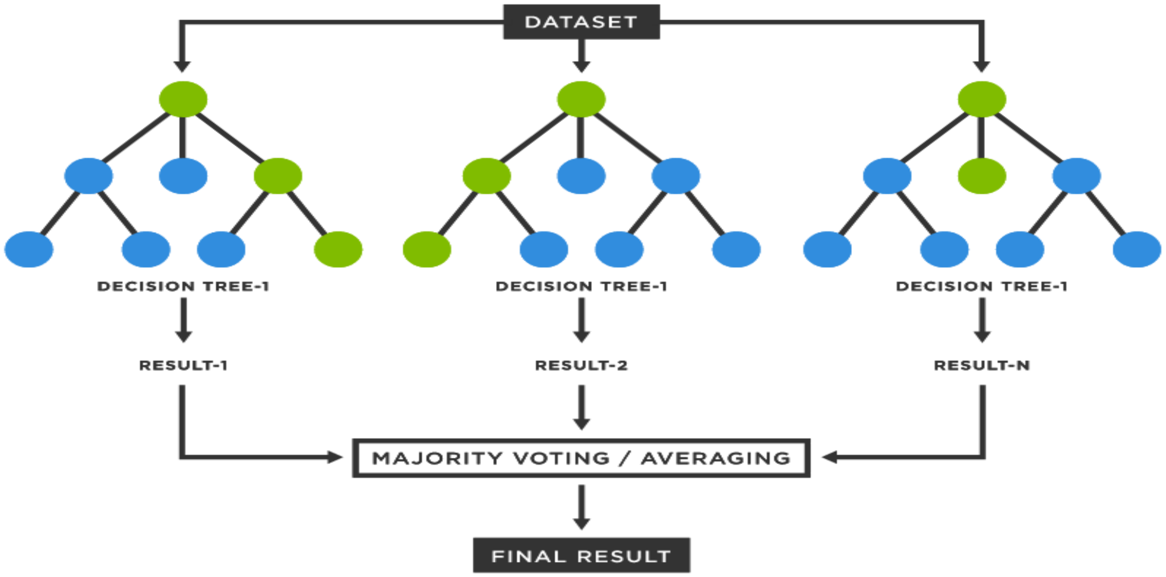
2.4 Decision Tree

To implement the Decision Tree algorithm in C/C++, a background study is essential. This involves understanding the fundamentals of decision trees, including their structure, learning process, and classification abilities. Additionally, familiarity with data preprocessing techniques, entropy calculation, and attribute selection measures is crucial. A solid grasp of C/C++ programming concepts, data structures, and algorithms is also necessary for efficient implementation.



2.5 Random Forest

The background study to implement the Random Forest algorithm in C/C++ involves understanding decision trees, ensemble learning, and the concept of bagging. It requires knowledge of data preprocessing techniques, feature selection, and handling categorical data. Additionally, familiarity with tree construction algorithms, such as ID3 or CART, is crucial for building the base classifiers in the Random Forest ensemble.



3. Project Description

In this endeavor, I aim to implement and explore the power of popular algorithms such as K-Nearest Neighbors (KNN), K-means, Decision Trees, Logistic Regression, and Random Forests.

Throughout this project, I will dive into the intricacies of each algorithm, understand their theoretical foundations, and implement them using C/C++. My focus lies in developing robust and efficient code that can handle diverse datasets and deliver accurate results.

By implementing these algorithms, I gained insights into classification, clustering, and regression tasks. We will also explore techniques for data preprocessing, feature selection, model evaluation, and hyperparameter tuning to optimize the performance of our algorithms.

Join us on this exciting journey as we unlock the potential of KNN, K-means, Decision Trees, Logistic Regression, and Random Forests. Through hands-on implementation and experimentation, we will gain a deeper understanding of their inner workings and their applications in various real-world scenarios.

Whether you are a beginner exploring the world of machine learning or an experienced programmer seeking to enhance your skills, this project will provide you with valuable knowledge and practical experience in implementing these powerful algorithms using C/C++. Let's embark on this exciting machine learning adventure together!

4. Implementation Details

Here are the mathematical equations associated with each of the machine learning algorithms mentioned above:

K-Nearest Neighbors (KNN): No specific equation, but the algorithm follows the principle of finding the majority class among the k nearest neighbors of a data point based on a distance metric (e.g., Euclidean distance).

K-means:

- Objective function: Minimize the sum of squared distances between data points and their corresponding cluster centroids.
  Equation for updating cluster centroids: $\mu_i = (1/|C_i|) \sum_{\text{rie}^{\text{l}}\text{e□e□□e□ xer□}}$
- (where μ_i represents the centroid of cluster C_i, $|C_i|$ is the number of data points in cluster C_i, and $_{\text{xer□}}$ represents a data point in cluster C_i)

Decision Tree:

- (where p_i is the probability of class i in a given node)
  Entropy (for classification):$\text{Entropy}(p) = - \sum_i p_i \log_2(p_i)$
- (where p_i is the probability of class i in a given node)
  Information Gain (for determining the best split):
  $\text{InformationGain}(D, A) = \text{Entropy}(D) - \sum_i (|D_i| / |D|) \text{ Entropy}(D_i)$
- (where D is the parent node, A is a candidate feature, D_i represents the subset of data points for which feature A takes value i)
  Random Forest:

- No specific equations, but Random Forest is an ensemble algorithm that combines multiple decision trees using techniques such as bagging and random feature selection.

Logistic Regression:

- Sigmoid function (Logistic function):$\sigma(z) = 1 / (1 + e^{(-z)})$
  (where z is the linear combination of feature values and their corresponding weights)
- Hypothesis function:$h(x) = \sigma(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + ... + \theta□x□)$
  (where h(x) represents the predicted probability that a given example x belongs to the positive class, and $\theta_0, \theta_1, \theta_2, ..., \theta□$ are the model parameters)
- Log Loss (cost function for binary classification):$J(\theta) = - (1/m) \sum(y \log(h(x)) + (1-y) \log(1-h(x)))$
  (where J(θ) represents the cost, m is the number of training examples, y is the true label, and h(x) is the predicted probability)

These equations provide a glimpse into the mathematical foundations of these machine learning algorithms.

5. User Manual

Here I tried to maintain a super easy user interface.I wrote all the functions in different header file.And then added them in a single file called **“Main.cpp”**.

Here I have written clean code which is understandable without any description or comments(a code smell).Here you will just write down which algorithm you want to run.I took a string **“s”.**If you put **s = “Any name of the five algorithms”** then you will be given the accuracy of this algorithm and then you can run the test data as well.

For example, you have given s = Knn, then you will be able to see the super accuracy of this algorithm.After that if you want to test any data and try to use it in different scenarios then just comment out the code which is used for testing data.

Here is the short description of the **input** of each algorithm:

Knn : Here you can give just a point (x,y) of two dimensional graph and its class following the same row.And in testing data if you put any point(x,y) you will be given the class of this point.here is some sample input :

```
0.12,0.76,1
0.46,0.87,1
```

K Means: This will also take some point as input.There is no class following the same row.You will give some point to train it and then you will get how many clusters are there.
Sample input :

```
1.1 2.3
4.5 3.2
6.7 8.9
```

Logistic regression : There are 4 features in this algorithm.I used iris dataset where 4 features are Sepal Length,Sepal Width,Petal Length,Petal Width.Also a class followed by each row will be the training data as well.('1' means its a iris flower,'0' means no)
Sample input

```
1,5.1,3.5,1.4,0.2,0
2,4.9,3.0,1.4,0.2,0
3,4.7,3.2,1.3,0.2,0
```

Decision Tree and Random Forest : Here as input you can take a maximum of 4 features and a class by following the same row.And every feature and the class will be in string format.

Sample input

```
Sunny Hot High Weak No
Sunny Hot High Strong No
Overcast Hot High Weak Yes
```

Here is the Main function’s code which can add all of my algorithms in a single point :

```
    string s; cin >> s;
    if (s == "knn")
    {
        knn_algo();
    }
    else if (s == "kmeans")
    {
        kmeans_algo();
    }
    else if (s == "logistic")
    {
        logistic_Regression();
    }
    else if (s == "random")
    {
        random_forest();

    }
    else if (s == "decision")
    {
        freopen("decision_random.txt", "r", stdin);
    }
```

6. Challenges

As I didn't know a little bit about machine learning so at the beginning I faced some difficulties and these were challenging as well.I numbered some of them according to the best of my knowledge.

1.Searched a book which can help me to give the basic concepts of ML.Then I bought “Machine learning Algorithm” by Tahmid Rafi.It helped me a lot.
2.As there is no code in c/c++ in online so I had to take help from some of my seniors.
3.I wrote almost all of the codes but didn’t check the accuracy.Then I learned again about k-folds cross validation and checked the accuracy of each of them.

These three are the key challenges actually.I faced many challenges related to this project throughout the semester. **Such as : Data preprocessing,Algorithm complexity,syntax of code,Handling categorical data,Feature selection and dimensionality reduction**.But at the end alhamdulillah I made it complete.

7. Conclusion

Even before the start of this semester, I was excited with the thought that SPL-1 is going to be my first solo project as a Software Engineering student. I picked up a field which is highly related to mathematics as it's one of my preferred subjects since high school days. It was a bit challenging at first to do background research about all the unknown stuff.I now fulfilled all of my scopes of the project as I mentioned in mid presentation and probably added a few more functionalities while doing it . This project has introduced me to a lot of new sides of programming. I have plans to continue working on this project. I look forward to adding more varieties of methods regarding ML.Hope that I’ll be able to add these extras in the near future.

This project can be useful for ML beginners to learn the basics of ML algorithms in an easier way.As for me, I believe that I have developed myself more and now I can actually think like a developer. I know how to manipulate, implement and design my coding now better than before. Overall, it was a great learning experience and surely it’ll work as my motivation for my upcoming projects.

8. References

Logistic regression with iris dataset- https://www.codingninjas.com/codestudio/library/applying-logistic-regression-on-iris-dataset(Last accessed on 05/04/23)

Decision Tree with playing tennis dataset - https://nulpointerexception.com/2017/12/16/a-tutorial-to-understand-decision-tree-id3-learning-algorithm/#:~:text=You%20might%20have%20seen%20many,yes%2C%20you%20may%20play%20tennis.(Last accessed on 14/03/23)

Random Forest - https://en.wikipedia.org/wiki/Random_forest(Last accessed on 20/03/23)