*Implementation of KNN Classifier for Sequence Prediction*

Zaka Ahmed   
zaka.ahmed@stud.fra-uas.de

Muhammad Haris  
Muhammad.Haris@fra-uas.de

*Abstract*—Machine learning comes out as an advanced and significant technology, evolving in our daily lives. The focus of Machine Learning is to enable computers to learn from the data and make predictions or decisions based on that data, enabling informed decision-making without the need for extra instruction or explicit programming for the system to perform tasks better. The sectors in which machine learning is involved include education, telecommunication, retail, research and development, finance, healthcare, and transportation through data-driven insights. Machine learning has three basic types: Supervised Learning, Unsupervised Learning, and Reinforcement learning. In the domain of supervised machine learning, a variety of classifiers abound, including decision trees, support vector machines, Naive Bayes, and K-nearest neighbors (KNN), each tailored to address specific data analysis challenges. Among the numerous machine learning algorithms, the K-Nearest Neighbors (KNN) classifier is the simplest and most effective for classification problems. This paper presents the implementation of the k- k-nearest neighbor (KNN) machine learning algorithm for predicting outcome variables based on input variables. Leveraging the capabilities of Hierarchical Temporal Memory (HTM) for learning complex temporal patterns, our study focuses on predicting types of sequences: even number sequences, odd number sequences, and decimal number sequences. We integrate the KNN model with the Neocortex API to efficiently classify sequences. The KNN model receives input data in the form of Sparse Distributed Representations (SDR) from HTM. We construct a dataset comprising multiple sequence SDRs, each with varying values within a defined threshold. The KNN model processes a stream of sequence SDRs, with the dataset split into 70% training data and 30% testing data. During testing, the model accurately classifies sequences, achieving a 90.9% accuracy rate with some SDR testing data, and consistently predicts matches with 100% accuracy in most cases. The paper discusses the KNN design procedure, challenges encountered, and potential enhancements to further improve model accuracy.

Keywords— Machine Learning, Hierarchical Temporal Memory, K-Nearest Neighbors, Sequence Classification, Integration, Neocortex API, Accuracy Enhancement.

# **Introduction**

KNN was first developed by Joseph Hodges and Evelyn Fix in the year 1951[1] where further development and we can say modification were led by Thomas Cover [2], in statistics the concept of k-nearest neighbors’ algorithm(k-NN) is involved in the non-parametric supervised learning method. KNN is commonly used for regression and classification. The input consists of K-neighbor training examples in both regression and classification. Remember the output depends on whether the use case is either regression or classification of K-NN.

## Regression

The main difference between classification and regression is that in regression, the output that we get will become the object’s property value. The value is the sum of the nearest neighbor’s values averaged together. If k=1, the output is assigned from that single nearest neighbor.

## Classification

The primary difference between the classifier and regression outputs is that in the former case, the output is the class membership. In classification, the object is classified based on the votes of its nearest neighbors. If k = 1, then the object will be in the class of that single nearest neighbor. The simple function of the KNN model is to predict the target class label. In other words, the class label is often described as a majority voting. The most common terms are technically considered” plurality voting” and” majority vote” The term” majority voting” means the majority needs to be greater the 50% for making decisions. The classification problems with only two classes, like binary predictions, there is always a majority. A majority vote is also automatically a plurality vote. We don’t require multi-class settings to make predictions via KNN in multi-class settings.

# Methods

This section should describe your work in detail. Here you can use references to your work and external sources. This section includes multiple subsections including a description of KNN in detail, The second section tries to focus on the Theoretical background including how the values determined the accuracy and complexity of the classifier. The last section of this section is regarding Hierarchical Temporal Memory (HTM).

## Literature Review

The K-Nearest Neighbors (KNN) algorithm is a widely used non-parametric classification method in machine learning. KNN relies on instance-based learning, in which the similarity of newly added data points to previously labeled data points is used to make predictions. In finance, KNN is used for credit scoring, stock analysis, and fraud detection. In the healthcare sector, KNN is used for patient health monitoring and drug suggestions. If I talk about marketing professionally KNN is used for product suggestion, market trend analysis, and customer ad suggestions. In most cases, KNN is used for determining specific patterns and tasks like face recognition or image classification. Researchers have proposed various enhancements to the traditional KNN algorithm which can improve scalability and performance.

The weight adjustment algorithm proposed by Han EH. proposed assigning weights to the nearest neighbors based on their distance from the respective point.[11][8]. The assigned weights distinguish, how much the weights influence the classification method. In this way, high weights will be assigned to the ones who are closer neighbors, so it gives more priority to similar instances while performing classification. This technique is useful where the dataset has many features, some of which can be considered unnecessary, but it has a high cost in the context of computational cost.

Zhang et al. have proposed to adjust the value of K based on the local density of data points [3]. This adaptive KNN algorithm dynamically selects the optimal value of K for each query point, leading to more robust predictions. And shows that the adaptive algorithm outperforms many other traditional KNN algorithms. The other approach can be a locally adopted KNN algorithm. It chooses the optimal value of K for classifying an input by analyzing the outcomes of cross-validation computations within the local neighborhood of the unlabeled data point.[12] The approach defined by Song Yang is to introduce two input parameters. As we know determining the correct value of K depends on the characteristics of the dataset, the selection of the correct parameter for various applications is a challenge. Song Yang suggests introducing a novel metric that assesses the informativeness of objects to be classified, with informativeness quantifying the significance of data points. Two parameters will be K and I as input. The class is determined based on the majority class of the most informative training examples.[10] Whereas some have proposed to integrate KNN with dimensionality reduction methods such as principal component analysis(PCA) to improve computational efficiency [5]. This combination is an example of a hybrid approach. Whereas others have combined KNN with ensemble methods such as random forests to enhance predictive accuracy [6]. These hybrid approaches enhance the performance as compared to standalone KNN.

## K-Nearest Neighbors Parameters and Matrix

The K-nearest was first used by the US Air Force to execute characteristics analysis. Different parameters in the KNN classifier play an important role in algorithm designs including distance matrix, K-Value selection, and voting.

1. **Computing Distance Matrix**

If we try to revise, the k-nearest neighbor algorithm’s main objective is the identification of the nearest neighbors around the input. The step will be given a label or name class to that specific point. The first thing is determining the distance metrics. To find which class (data point) is nearest to the input data, to do so we will calculate the distance between the data points and query point. We get assistance to decide in which regions the input point belongs. The distance metrics can be either Manhattan distance or any other approach. The first thing is to identify the k- k-nearest neighbors and then the number of its k-neatest neighbors. The most famous techniques are discussed below:

1. **Euclidean distance (p=2)**

In the early 300s Era before, mathematicians of Greece introduced Euclid while finding the difference between distance and angle. Still, Euclid is the most commonly used of distance. From that early, till yet Euclid is widely and applies in two- or three-dimensions space. The main method is the root of square distances between two coordinates of a pair of objects. Then there is a square root of the sum of squares of the differences between the corresponding values. [45]

A graph with lines and points

Description automatically generated

Fig 1: Euclidean Distance [50]

As from Fig. 1, we have the , and , we can see that , and are allocated into the two-dimensional space. If we try to build a right-angled triangle and draw a hypotenuse straight line between two points(d). The other two sides of a right-angle triangle are base and altitude which will be and . So there the hypotenuse(d) is our Euclidean distance which is between ,, and . As this is only a straight line we will use Pythagorean theorem. The distance between ,, and , would be , and .

Euclidean distance = d(x, y) =

* *yi* are the coordinates of one point.
* *xi* are the coordinates of another point.
* d is the distance between (*x*1*, y*1)*, and,* (*x*2*, y*2).

Moreover, to find the Euclidean distance between, *yn, and xn* the Euclidean distance is (*xn − yn*)2 in n-dimensional space. Euclidean is limited to real-valued vectors and is most used for distance measures. The straight line between the input point and the other point under measurement is found using the expression above.

Euclidean distance is the shortest distance between two points means the straight line between the source and destination whereas the Manhattan distance is the sum of all the real distances between the source and destination. And each always be a straight line. The reason behind Euclidean distance used in KNN is that it provides a simple and intuitive measure of similarity between data points in a multi-dimensional space. One of the main reasons is that Euclidean distance in KNN is its computational efficiency and simplicity.[45] Euclidean distance metric works well for continuous features and is suitable for data represented in a high-dimensional space.

Euclidean Distance is largely used in machine learning learning algorithms such as K-Nearest(KNN) where it is used to measure the similarity between two data points. Spatial Analysis for measuring distance between geographical locations.[45] As well as, in robotics for obstacle avoidance and path planning. It is also used in the image process for comparing images based on pixel values. Due to the coordinate system, it is robust to transformations of the data. It is the most widely adopted approach in machine learning applications.

1. **Manhattan distance**

In Manhattan distance, the absolute value is measured between two points. Manhattan is widely used for resolving problems related to geometry. We can say it is an ordinary distance between two points. Manhattan distance is also one of the popular and dominant distance metrics. The most common example is the Uber app visualization with a grid and navigation via streets from one address to another illustration. Manhattan is widely used in cluster analysis. The k-Means clustering algorithm is the common example where the Manhattan distance is used.[46] The second name of Manhattan Distance is city Block Distance. The easiest method to figure out the distance is to move from one point to the other between two locations by moving horizontally and then vertically, rather than straight forward.[46][56] It only needs to subtract instead of performing complicated calculations that why it is one of the simplest methods of calculating the distance.

A graph with a line drawn on it

Description automatically generated

Figure 2: Manhattan Distance [49]

Manhattan Distance = d(x, y) =

As shown[56], it is the root of the squared difference between the coordinates of two objects. We only need to subtract two points instead of performing complex tasks.

# Results

This Part of the text describes results of your works. There can only be mentioned references, MUST point back to Methods and Intro chapter. No more external references.

Code examples must be provided to demonstrate how to use the algorithm/module. Provide a reference to more unit tests, which show the same in more detail. Also provide all diagrams with comments and reference to unit tests, which generate diagrams.

# Discussion

Conclusion of your work should be precise and concise. How was the project, what is done, what is the result... There can be discussion on further work and direction.

# Ease of Use

## Selecting a Template (Heading 2)

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size. If you are using US letter-sized paper, please close this file and download the Microsoft Word, Letter file.

## Maintaining the Integrity of the Specifications

The template is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin in this template measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

# Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections A-D below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not use hard tabs, and limit use of hard returns to only one return at the end of a paragraph. Do not add any kind of pagination anywhere in the paper. Do not number text heads-the template will do that for you.

## Abbreviations and Acronyms

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, sc, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

## Units

* Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
* Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
* Do not mix complete spellings and abbreviations of units: “Wb/m2” or “webers per square meter”, not “webers/m2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.

may use the solidus ( / ), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

*a**b* 

Note that the equation is centered using a center tab stop. Be sure that the symbols in your equation have been defined before or immediately following the equation. Use “(1)”, not “Eq. (1)” or “equation (1)”, except at the beginning of a sentence: “Equation (1) is . . .”

## Some Common Mistakes

* The word “data” is plural, not singular.
* The subscript for the permeability of vacuum **0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter “o”.
* In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
* A graph within a graph is an “inset”, not an “insert”. The word alternatively is preferred to the word “alternately” (unless you really mean something that alternates).
* Do not use the word “essentially” to mean “approximately” or “effectively”.
* In your paper title, if the words “that uses” can accurately replace the word “using”, capitalize the “u”; if not, keep using lower-cased.
* Be aware of the different meanings of the homophones “affect” and “effect”, “complement” and “compliment”, “discreet” and “discrete”, “principal” and “principle”.
* Do not confuse “imply” and “infer”.
* The prefix “non” is not a word; it should be joined to the word it modifies, usually without a hyphen.
* There is no period after the “et” in the Latin abbreviation “et al.”.
* The abbreviation “i.e.” means “that is”, and the abbreviation “e.g.” means “for example”.

An excellent style manual for science writers is [7].

# Using the Template

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

## Authors and Affiliations

**The template is designed for, but not limited to, three authors.** A minimum of one author is required for all report articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

### For papers with more than three authors: Add author names horizontally, moving to a third row if needed for more than 8 authors.

### For papers with less than three authors: To change the default, adjust the template as follows.

#### Selection: Highlight all author and affiliation lines.

#### Change number of columns: Select the Columns icon from the MS Word Standard toolbar and then select the correct number of columns from the selection palette.

#### Deletion: Delete the author and affiliation lines for the extra authors.

## Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is “Heading 5”. Use “figure caption” for your Figure captions, and “table head” for your table title. Run-in heads, such as “Abstract”, will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced. Styles named “Heading 1”, “Heading 2”, “Heading 3”, and “Heading 4” are prescribed.

## Figures and Tables

For adding object other than text (tables, equations, graphs, figures, code…), **there must be at least one cross reference** to it. Figure 3 is an example

#### Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation “Fig. 1”, even at the beginning of a sentence.

1. Table Type Styles

| Table Head | Table Column Head | | |
| --- | --- | --- | --- |
| Table column subhead | Subhead | Subhead |
| copy | More table copya |  |  |

1. Sample of a Table footnote. (*Table footnote*)



Figure 3 Example Figure Caption

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity “Magnetization”, or “Magnetization, M”, not just “M”. If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write “Magnetization (A/m)” or “Magnetization {A[m(1)]}”, not just “A/m”. Do not label axes with a ratio of quantities and units. For example, write “Temperature (K)”, not “Temperature/K”.

## Code References:

Referencing Code in your text should be avoided unless necessary. In such cases it can be inserted as a listing as shown in **Error! Reference source not found.**

Listing 1 Code Reference Example

Console.WriteLine(“Referencing code”, var);

// using tab can be replaced with 4 spaces

Do not pass code as image. When referring to variable in **Error! Reference source not found.**, italics should be used for example *var.* Code flows and logic should be presented better as Graph or Diagram instead of words.

Code Block which is too big to put in the textbox can be reference as Listing 2.

Listing 2 Unit Test [EncodeDateTimeTest](https://github.com/ddobric/neocortexapi/blob/0348ffb99739ddf8c8c3a875f8162a18073938ca/source/UnitTestsProject/EncoderTests/DateTimeEncoderExperimentalTests.cs#L34-L49)

public void EncodeDateTimeTest(int w, double r, …)

{

…

DateTimeEncoderExperimental encoder = new…

var result = encoder.Encode(input);

…

Assert.IsTrue(result.SequenceEqual(expected…

}

##### Acknowledgment *(Heading 5)*

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

##### References

The template will number citations consecutively within brackets [1]. The sentence punctuation follows the bracket [2]. Refer simply to the reference number, as in [3]—do not use “Ref. [3]” or “reference [3]” except at the beginning of a sentence: “Reference [3] was the first ...”

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors’ names; do not use “et al.”. Papers that have not been published, even if they have been submitted for publication, should be cited as “unpublished” [4]. Papers that have been accepted for publication should be cited as “in press” [5]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

1. G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. *(references)*
2. J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
3. I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
4. K. Elissa, “Title of paper if known,” unpublished.
5. R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.
6. Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].
7. M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

**This report template contains guidance text for composing and formatting technical reports. Please ensure that all template text is removed from your report prior to submission to the examination office. Failure to remove template text from your paper may result in your paper being degraded.**

We suggest that you use a text box to insert a graphic (which is ideally a 300 dpi TIFF or EPS file, with all fonts embedded) because, in an MSW document, this method is somewhat more stable than directly inserting a picture.

To have non-visible rules on your frame, use the MSWord “Format” pull-down menu, select Text Box > Colors and Lines to choose No Fill and No Line.