**CS469 Data Structures and Algorithms**

**HOS10 K-Nearest Neighbors**

06/05/2023 Reviewed by Christopher Sharp

03/04/2024 Reviewed by Anh Nguyen

09/29/2024 Reviewed by Shahid Khan

School of Technology and Computing (STC) @ City University of Seattle (CityU)

**Before You Start**

* The document’s examples are written in Python. Please finish the Python tutorial in the Module00 folder before you start the assignment.
* Some steps are not explained in the tutorial**.** If you are not sure what to do:
  1. Consult the resources listed below.
  2. If you cannot solve the problem after a few tries, ask a TA for help.

**Learning Outcomes**

Students will be able to:

* Understanding the process of splitting a dataset into training and testing sets for model evaluation.
* Gaining hands-on experience in implementing the K-nearest neighbors algorithm using the scikit-learn library.
* Experimenting with different values of the 'n\_neighbors' parameter to observe the impact on model performance.
* Analyzing the accuracy of the classifier with different parameter settings to determine the optimal 'n\_neighbors' value.
* Observing the predicted labels and comparing them with the true labels to evaluate the performance of the classifier.
* Gaining insights into the importance of parameter selection and its influence on the accuracy of a machine learning model.

**Resources**

* Python Tutor. <https://pythontutor.com/visualize.html>
* MachineLearning - KNN using scikit-learn. <https://towardsdatascience.com/knn-using-scikit-learn-c6bed765be75>
* Guttag, J. (2017). *Classification*. MIT OpenCourseWare.
* Starmer, J. [StatQuest with Josh Starmer]. (2017, June 26). *StatQuest: K-nearest neighbors, Clearly Explained* [Video]. YouTube. https://youtu.be/HVXime0nQeI

# Introduction

K Nearest Neighbors (KNN) is a supervised learning algorithm mainly used for classification and regression.

The principle of KNN is simple: memorize the training data and find the closest data point to the unknown data point, then assign the label of the closest data point to the unknown data point.

For example, the data points of training data can be seen below. The data points are separated into two classes: red and black. The x point is a new point – we need to predict the class of x.

The KNN algorithm will find the closest point to x which is the red point at the right side of x. The KNN algorithm then predicts the class of x is red.

A group of black and red dots

Description automatically generated with low confidence

Figure 1. KNN classification (Guttag, 2017)

KNN finds the closest point to x by calculating the Euclidean distance between x and each point.

## Formula

A picture containing font, white, handwriting, diagram

Description automatically generated

p, q = two points in Euclidean n-space  
qi, pi = Euclidean vectors, starting from the origin of the space (initial point)   
n = n-space

## Setting up the environment and implementing KNN:

1. Open VS Code
2. You should be in your repository folder
3. Create a new file named **knn\_hos.py**
4. Import the necessary libraries:

* In the new Python file, add the following import statements at the beginning of the file:

A picture containing text, screenshot, font

Description automatically generated

1. Implementation:

Type the following code in your file,

A screen shot of a computer program

Description automatically generated with low confidence

1. Save the file:

Click on "File" and select "Save" to save the Python file.

1. Run the code:

* Open the integrated terminal in VS Code by clicking on "View" in the menu bar and selecting "Terminal" or using the shortcut (Ctrl+`).
* In the terminal, navigate to the directory where you saved the Python file using the cd command (e.g., cd Module10).
* Execute the Python file by running the command “**python [filename].py”** in the terminal.
* The output will be displayed in the terminal, showing the accuracy and the predicted results for each value of 'n\_neighbors'.

**Example:**

A picture containing text, screenshot, font, black and white

Description automatically generated

# Push Your Work to GitHub

# A black background with white numbers and letters Description automatically generated

**Save output screenshots in this document and save this document as a PDF file in the module folder.**

Open a terminal on visual studio code and make sure you’re in the repository folder. (i.e: hos01\_courseName\_GitHubUserName)

**Type the following command to upload your work**:

>>> git add .

>>> git commit -m "Submission for HOS10 - YourName"

>>> git push origin master