university of groningen / faculty of science and engineering / artificial intelligence

Experience based teaching of AI in Dutch high schools

Bachelor's Project Thesis

Max Stevens, s3948277, m.g.stevens.2@student.rug.nl
Supervisor: J.D. Cárdenas-Cartagena

Abstract:
As the societal impact of AI grows, it becomes more important that high school students learn about this technology. At the same time, there is a reported gap for this type of course in the Dutch high school curriculum. To address this gap, we explore designing teaching material aimed at teachers in Dutch high schools with a general math background. The goal of the material is to teach students about using the nearest neighbour algorithm for social media recommendations. This algorithm is chosen for its close relation to the mathematics taught in Dutch high schools and for its wide range of practical use cases, both of which are intended to increase the effectiveness of the teachings.

1 Introduction

The societal impact of artificial intelligence (AI) is growing. To keep understanding this rapidly evolving world, and to be able to critically evaluate the usage of these AI algorithms, it is important that the next generation learns about the workings and drawbacks of AI. To move toward these goals, high school students must be offered courses in AI. However, due to the existing shortage of informatics teachers in the Netherlands (Ministerie van Onderwijs, Cultuur en Wetenschap, 2021), these courses are not regularly taught in Dutch high schools. Another problem that arises is that, as with other research disciplines, AI is a complex field. To teach this material to high school students, the theory must be distilled into material that the students can understand and learn from. The aim of this research is to create effective teaching materials that high school teachers with no experience in the field of AI can use to teach their students about AI.

In this research the design thinking methodology is used to explore different teaching approaches. The first one is a proof-of-concept (POC) implementation of the nearest neighbour algorithm (NN) in the online programming environment Scratch. Based on this POC, a demo of a social network is created to showcase a real-life application of the NN algorithm.

2 Related work

2.1 AI in high schools

Martins & Gresse Von Wangenheim (2023) conducted a review of studies aimed at teaching AI to high school students. They found that a wide range of topics in AI could be effectively taught to high school students. Mobasher et al. (2019) effectively taught students different algorithms, including decision trees and k-nearest neighbours, by using a lab activity where the students solved different data problems using the algorithms.

In another study, Park & Kwon (2023) created an 8 week, 16 hour long AI programme for students in Korean middle schools. This programme included data analytics, physical computing, AI ethics and machine learning and used both instruction and lab assignments to teach the topics. The researchers concluded that the students significantly improved their AI competencies (Park & Kwon, 2023).

2.2 Teaching NN

The scope of this research is limited to teaching NN to high school students.

Mariescu-Istodor & Jormanainen (2019) explored specifically the teaching of this algorithm. They did this by creating an activity where the students implemented an image classification system based on the NN algorithm. For this activity the students were gradually introduced to the different facets of the application. They were given a bare-bones system on top of which they had to implement the classification application. This research has shown that AI can be taught effectively to high school students in Romania. However, these students already had existing programming experience, so teaching about programming was not one of the goals of the research. Our research tries to create teaching methods that do not require any programming experience, to tailor to the Dutch students.

3 Theoretical Framework

3.1 Nearest neighbour

The nearest neighbour algorithm is a foundational algorithm in the field of Artificial Intelligence (Peterson, 2009). It can be applied to a wide range of scenarios, where it can achieve good performance on classification tasks. The algorithm is mostly used for classification tasks, where classes are assigned based on the data points that are closest to the data point that is being classified. The class that is the most prevalent in the neighbouring points will be the prediction for the unknown data point. In this scenario, the algorithm is called k-nearest neighbours, where k denotes the amount of neighbours considered for the prediction (Peterson, 2009).

NN can also be applied to other tasks, for example to recommendation systems. By ranking the neighbouring points, a list of recommendations can be created where the closest point is also the best recommendation. The assumption here is that the more similar two points are, the better fit they are for each other (Adeniyi et al., 2016).

To determine how similar two points are, a distinction between quantitative (numerical) and qualitative (categorical) properties is made. Different formulas are needed to compute the distances between these types of points. The distance between the quantitative properties of two multidimensional points p and q (d_e) can be calculated using the (weighted) euclidean distance metric:

$$d_e(p,q) = \sqrt{\sum_{i=1}^{n} w_i \cdot |p_i - q_i|^2}$$

where n denotes the amount of quantitative properties in the points, and w denotes a weight for each property. The weights can changed based on for example the absolute values of the properties. The distance between the qualitative properties of two points (d_h) can be calculated using the hamming distance:

$$d_h = 1 - \frac{c_{\text{equal}}}{c_t}$$

where c_{equal} denotes the amount of properties that have the same value between the points, and c_t denotes the total amount of qualitative properties in the points. In order to calculate the total distance between the points (d_t), the sum of the distances is used:

$$d_t = d_e + d_h$$

This formula does not account for a potential difference in the amount of quantitative and qualitative properties. These proportions can be added to the previous formula to calculate the weighted distance $d_t^{'}$:

$$d_t^{'} = d_e \cdot \frac{c_e}{c_t} + d_h \cdot \frac{c_h}{c_t}$$

where c_e counts the amount of quantitative properties, and c_h the amount of qualitative properties.

4 Methodology

The goal of this research is to explore multiple ways of teaching AI in Dutch high schools. This is done by the creation of teaching materials.

4.1 Design Thinking

The design thinking approach was used to design the material (Razzouk & Shute, 2012). The approach consists of the steps empathize, define, ideate, prototype and test. This project uses the define, ideate, and prototype steps, since no user interviews and tests were performed.

Design thinking is useful as it provides a systematic approach to an often unstructured creative process. The goal of the method is to facilitate multiple iterations in this creative design process, without getting stuck on a single idea or approach.

4.2 Designing interactive teaching materials

The goal of this project was to create a teaching guide and complementary learning activity which could be used by teachers in Dutch high schools to teach about AI. An option that was considered early in the project was to create this material in Scratch. Scratch is an online programming environment developed by MIT aimed at students without any prior programming experience. Since we observed that Dutch high school students almost never had this experience, Scratch would be a suitable environment. Next to this, existing research had already shown that it was possible to teach both explainable AI (Alonso, 2020) as well as K-means clustering (Estevez et al., 2019) to high school students using Scratch. NN was chosen as the algorithm to explore in the POC (see appendix D).

Based on the POC it was decided that the implementation of NN in Scratch did not meet its expectations of being understandable to students who lack programming experience. The implementation required multiple unintuitive workarounds to fit into the Scratch programming model, which made it more difficult to relate to the implementation to the mathematics behind NN.

The next iteration of the activity aimed to relate the mathematics taught in Dutch high schools to the mathematics behind the NN algorithm without touching on the implementation details of the algorithm. During a web-based activity students would be able to view the results of the algorithm and change different parameters in order to see how these would affect the outcome.

It was theorized that in order to increase the effectiveness of this activity, it had to resemble a familiar environment. Social networks were one of those environments. As the recommendation systems used on the social networks seemed to gain more media attention, this was picked as the subject to base the activity on.

4.3 Social network dataset

The social network uses a dataset of 40 characters containing 8 comparable properties (appendix C). The properties were selected to provide a balance between quantitative and qualitative properties, be realistic properties for characters in a social network, and to be of different levels of relevance when making recommendations. The characters itself were created by ChatGPT and then tweaked manually to ensure a distribution which offered interesting patterns for each character. This included giving some characters more friends than others, giving every character a best friend with similar properties and ensuring that every character has a direct network that contains at least some friends.

5 Results

5.1 NN in social network recommendations

The material that is used to teach NN in social media recommendations includes a teaching guide (appendix A) and a web application (appendix B). The target audience of the teaching guide are the teachers using this material to teach AI to their students. The first section gives them instructions how to explain the basics of NN to their students. When this is taught, they move on to the web application.

5.1.1 Explaining NN

The teaching guide begins with an analogy between the NN algorithm and how neighbours are determined in real life. Next it is explained where NN can be useful and what the concepts behind it are, including a connection between NN and euclidean distance. The guide also makes sure to explain the difference between qualitative (categorical) and quantitative (numerical) properties. To gauge the competency of the students after the explanation, sample questions are provided.

5.1.2 The social network activity

When the students understand the fundamentals of NN, an activity is proposed to further enhance their understanding of the algorithm. This activity uses a web application that shows how NN can be used when making friend recommendations in a social network. This application consists of a homepage, a comparison page and a network page (Stevens, 2024).

5.1.3 The homepage

The homepage is shown in figure B.1 of the appendix. On this page students find their own character and their properties on the left, and their list of recommended friends on the right. Each character has a profile picture and a unique name, so that they are recognizable throughout the application. The list of recommendations is sorted from best to worst. The distance from the recommendation to the user's character is also shown.

5.1.4 The comparison page

The comparison page is shown in figure B.2 of the appendix. This page compares the character of the user to the selected character. The properties of both characters are presented side-by-side. The aim is that by looking at the comparison pages for multiple characters, the students will learn which properties affect the recommendations.

5.1.5 The network page

The network page is shown in figure B.3 of the appendix. This page shows the network of the user represented as a graph. This is another way to look at the recommendations that are made. The network that is shown consists of all the users that have a distance smaller than the distance threshold. This threshold can be controlled by the student. By changing this threshold, students can observe how the amount of people in the network will either increase or decrease. The aim is that this measure illustrates how the distance metric is used to make recommendations.

5.2 Neighbourhood graphs

The neighbourhood graph (appendix B.3) shows relations between the current character and the other characters in the network. To avoid visual clutter in the graph, connections between other characters are omitted. When comparing the local network graphs of different characters, the amount of connections will vary. While this will be familiar to the students it also provides an insight into the way the algorithm picks recommendations. This can be a starting point for the students to think further about how the recommendations are made.

6 Conclusion

The teaching guide and complementary material take into account that programming classes are optional and offered only to a subset of Dutch high schools. This is reflected by the omission of any programming specific content. Instead, the material shows the NN algorithm from a mathematical viewpoint, where the level of mathematics discussed reflects the level of mathematics taught to high school students in the second half of their high school careers. This makes the material suitable for high school students in their 4th year and up.

In order to fit the activity into the mostly full Dutch teaching curriculum, the amount of time needed is kept low. The activity consists of one session which is expected to take around two hours. This is beneficial from a scheduling perspective, especially compared to courses with longer and/or more sessions.

A limitation of this research is that the material is not tested in actual classrooms, which means that the effectiveness of the material is not verified. This would be the next step for future research. When used to teach, it is expected that the material will increase a students' understanding of AI, in particular of the NN algorithm. They should also learn how these systems are used in their everyday lives, and how systems can sometimes make incorrect predictions. Directions for future research include an extension to other subjects in AI like ethics or deep learning. Increasing the scope of the demo to contain more interactive elements like a chatroom would also be interesting to explore.

References

Adeniyi, D., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using k-nearest neighbor (knn) classification method. *Applied Computing and Informatics*, *12*(1), 90-108. doi: 10.1016/j.aci.2014.10.001

Alonso, J. M. (2020, July). Teaching Explainable Artificial Intelligence to High School Students. *International Journal of Computational Intelligence Systems*, *13*(1), 974–987. doi: 10.2991/ijcis.d.200715.003

Estevez, J., Garate, G., & Graña, M. (2019, November). Gentle Introduction to Artificial Intelligence for High-School Students Using Scratch. *IEEE Access*, *7*, 179027–179036. doi: 10.1109/ACCESS.2019.2956136

Mariescu-Istodor, R., & Jormanainen, I. (2019, November). Machine Learning for High School Students. In *Proceedings of the 19th Koli Calling International Conference on Computing Education Research* (pp. 1–9). New York, NY, USA: Association for Computing Machinery. doi: 10.1145/3364510.3364520

Martins, R. M., & Gresse Von Wangenheim, C. (2023, September). Findings on Teaching Machine Learning in High School: A Ten - Year Systematic Literature Review. *Informatics in Education*, *22*(3), 421–440. doi: doi.org/10.15388/infedu.2023.18

Ministerie van Onderwijs, Cultuur en Wetenschap. (2021, December). *Trendrapportage Arbeidsmarkt Leraren po, vo en mbo 2021* (Tech. Rep.). Ministerie van Onderwijs, Cultuur en Wetenschap. Retrieved 2024-06-18, from `https://open.overheid.nl/documenten/ronl-580c98e3-5765-4d4a-a511-62dd568ee0f4/pdf`

Mobasher, B., Dettori, L., Raicu, D., Settimi, R., Sonboli, N., & Stettler, M. (2019, May). Data Science Summer Academy for Chicago Public School Students. *ACM SIGKDD Explorations Newsletter*, *21*(1), 49–52. doi: 10.1145/3331651.3331661

Moreno-León, J., & Robles, G. (2016, April). Code to learn with Scratch? A systematic literature review. In *2016 global engineering education conference (EDUCON)* (pp. 150–156). Abu Dhabi, United Arab Emirates: IEEE. doi: 10.1109/EDUCON.2016.7474546

Park, W., & Kwon, H. (2023, February). Implementing artificial intelligence education for middle school technology education in Republic of Korea. *International Journal of Technology and Design Education*. doi: 10.1007/s10798-023-09812-2

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883. doi: 10.4249/scholarpedia.1883

Razzouk, R., & Shute, V. (2012, September). What Is Design Thinking and Why Is It Important? *Review of Educational Research*, *82*(3), 330–348. doi: 10.3102/0034654312457429

Stevens, M. (2024). *Social network app.* `https://github.com/markusgeert/social-network-app`. GitHub. (This code repository contains the source code for the interactive social network app.)

A Teaching guide

A.1 Introduction

The goal of the activity is that students will learn about one of the foundational algorithms of Artificial Intelligence (AI), the nearest neighbour algorithm. This algorithm can be used in a wide-range of applications and achieve remarkably good results. Examples of applications are image classification (e.g. determining which animal is in an image), determining the ideal sales price for a home and social media recommendations.

Material offered to the teacher is the interactive online example of a social network. This fictitious network has some of the characteristics of networks in real life. In the example, the nearest neighbours algorithm is used to find the most likely friends for each person in the network. Students using the demo are able to explore everyone in the network, make comparisons between their character and others and are able to view relations between people based on the results of the algorithm as a social network graph.

A.2 Nearest Neighbour

The nearest neighbour algorithm is a relatively simple and intuitive algorithm in the space of AI. It lends its name from neighbours in real-life, where someone living close or next to you would be a neighbour, while someone living far away wouldn't. The nearest neighbour algorithm works with the same concept except that we don't say someone is a neighbour based on physical distance, but we determine a neighbour based on the distance between properties of the person.

Based on the distance between our datapoint and the neighbours, we can make predictions about unknowns of our datapoint. Let's say we want to determine a certain property of person, for example if they're from a city or from a rural area. We compute the distance between our person and the other people, from whom we know where they're from.

The question that logically arises from this is how to calculate a distance between properties? A distance between two houses is intuitive to see, while a distance between properties is more abstract. An important connection to make here is that calculating the distance between properties can be done in the same way a distance between two (physical) points can be calculated.

In order to understand this, we must first make distinction between two types of properties. We have categorical properties and we have numerical properties. Categorical properties are properties that have certain distinct categories, for example nationality or brand. Numerical properties on the other hand describe quantities, for example the amount of students in a class.

A good exercise would be to let the students classify the following properties into their respective categories. If the students struggle with understanding the difference, consider providing example values for each property.

- Age (numerical)
- Gender (categorical)
- Height (numerical)
- Favourite Food (categorical)
- Time needed to get to school (numerical)

The way we determine the distances for each of these types of properties is different. For numerical properties, we can calculate the distance d_e via the Pythagorean method:

$$d_e(p,q) = \sqrt{w_1 \cdot |p_1 - q_1|^2 + w_2 \cdot |p_2 - q_2|^2 + \cdots + w_x \cdot |p_x - q_x|^2}$$

where p and q are the points, and x is the total amount of properties in the points. The weights w_x can be added to account for the differences in property values, or to tweak the output of the computations. The following formula can be shown to the students if a more concrete example is desired:

$$\text{Euclidean distance} = \sqrt{\text{age diff.}^2 + \text{height diff.}^2 + \text{screentime diff.}^2}$$

To determine the distance between categorical properties we use the 'hamming distance'. This distance relies on the assumption that categorical properties can either be exactly the same, or totally different. The hamming distance is determined based on the amount of categorical properties with the same value between the points. When they're all different, the distance is the biggest (1) and while they're all the same the distance is the smallest (0).

To put it more concretely, we can calculate the hamming distance d_h using the following formula:

$$d_h = 1 - \frac{c_{\text{equal}}}{c_t}$$

where c_{equal} is the count of properties with the same value, and c_t is the total amount of categorical properties.

The final step is to combine the two distances. To keep it simple here, we can just sum the two distances to get the final distance d_t:

$$d_t = d_e + d_h$$

This approach does have its caveats, especially when the ratio of the two property types is not 1:1. However, for this demonstration the simple algorithm of summing the two works fine. Creating a better formula for the total distance can remain as a question for the students. One example for this would be weighing the distances based on the ratio's of the two property types:

$$d_t^{'} = d_e \cdot \frac{c_e}{c_t} + d_h \cdot \frac{c_h}{c_t}$$

where c_e and c_h contain the total amount of numerical and categorical properties respectively.

A.3 Social network activity

The activity aims to illustrate the concept and show how the nearest neighbour algorithm can be used in real-life scenarios. In the provided demonstration the algorithm is used to provide friend recommendations in a social network.

The activity can be started by showing a walk through of the network to the students. For reference, see the following screenshots:

- Starting page - figure B.1
- Comparison page - figure B.2
- Network page - figure B.3

The students can explore this network from the perspective of one of the people in the network. They can look at the list of recommended friends and compare themselves to individual people on that list. They do this by clicking on the people in the list.

When comparing themselves to another person in the network, the properties of the two are placed next to each other. In this overview, students are able to see where their characters are similar and where they are different. The goal is that by doing this with multiple characters, students gain a sense of how the distances between characters are calculated, and

When the students open the network, they are assigned a person in the network at random. They can explore the social network from this perspective. Everyone in the example network has different properties, including their age, their gender, their favourite food and their preferred music.

A.4 Concluding

When the activity is concluded, you can use the following questions to evaluate the effectiveness of the course, and to spark a conversation about AI in general:

- What limitations do you see with this algorithm?
- Did some properties have more influence in the recommendations than others? Do you know why?
- How logical did you find the recommendations the algorithm made?
- How would you improve the recommendation algorithm?
- Do you think we should be using more AI based algorithms?

In the end, the goal is to teach students more about the applications, workings and downsides of AI. To learn more about this topic, and in particular large language models like ChatGPT, you can refer to this excellent guide from Harvard: `https://aipedagogy.org/`.

B Activity screenshots

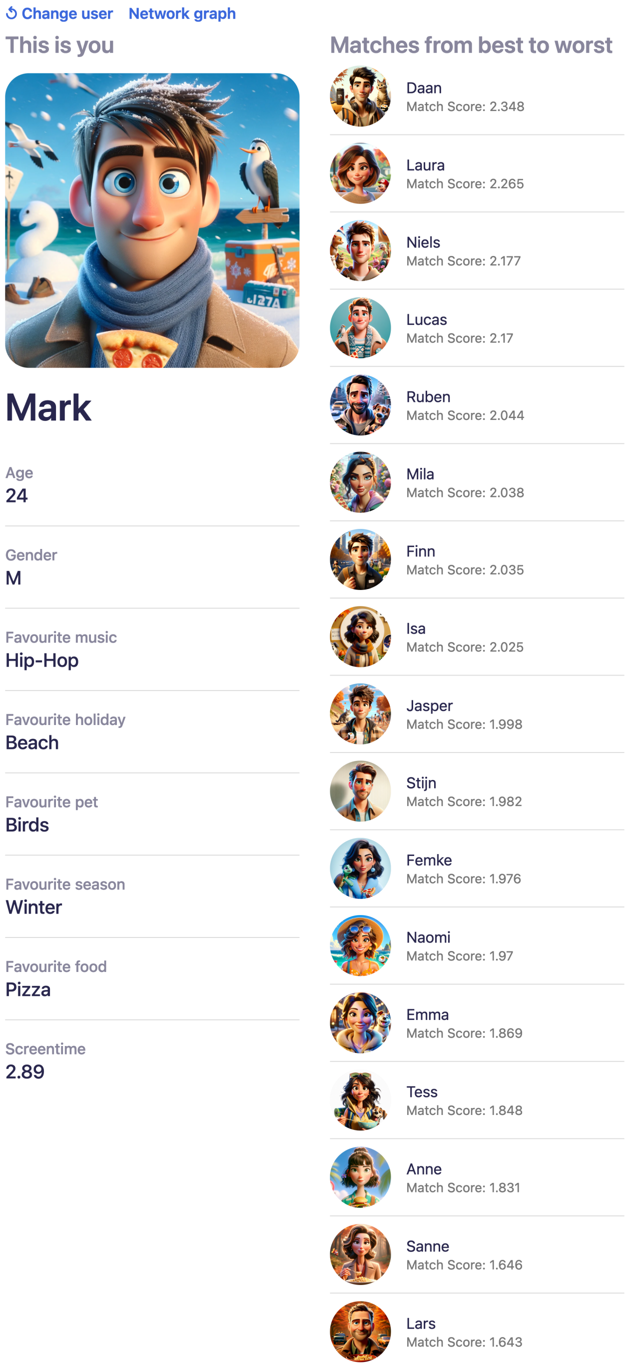

Figure B.1: The homepage of the activity

< Back

This is you	Recommended
	
Mark	**Laura**
Age 24	Age 33
Gender M	Gender V
Favourite music Hip-Hop	Favourite music Hip-Hop
Favourite holiday Beach	Favourite holiday Nature retreat
Favourite pet Birds	Favourite pet No pets
Favourite season Winter	Favourite season Autumn
Favourite food Pizza	Favourite food Pizza
Screentime 2.89	Screentime 2.72

Figure B.2: The comparison page of the activity

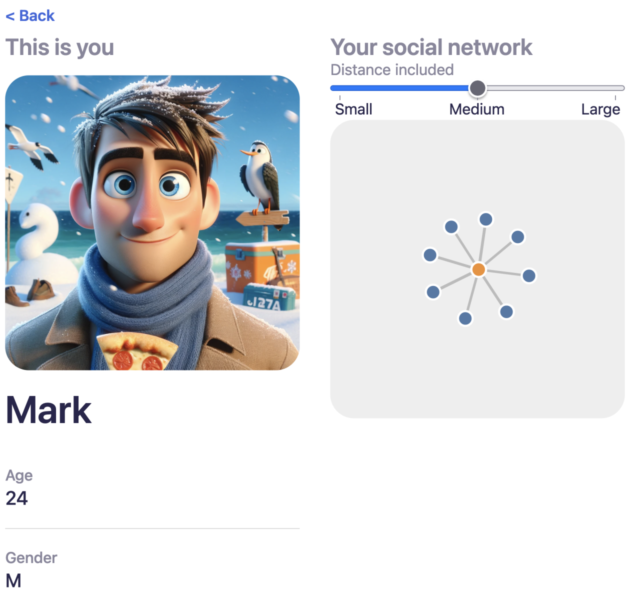

Figure B.3: The network page of the activity

C Dataset

Name	Age	Screentime hrs./day	Gender	Music	Pet	Fav. food	Fav. season	Fav. holiday
Anne	16	4.60	V	Indie	Cats	Burgers	Spring	Beach
Levi	16	6.20	M	Rock	Dogs	Burgers	Summer	Citytrip
Jasper	17	4.67	M	House	Cats	Sushi	Autumn	Beach
Bas	19	7.36	M	Hip-Hop	Fish	Pasta	Winter	Citytrip
Anouk	19	5.47	V	House	Cats	Sushi	Spring	Nature retreat
Niels	21	3.72	M	Pop	Cats	Chocolate	Spring	Hiking
Finn	22	4.22	M	Rock	No pets	Sushi	Autumn	Citytrip
Mark	24	2.89	M	Hip-Hop	Birds	Pizza	Winter	Beach
Max	25	7.40	M	Pop	No pets	Pizza	Summer	Beach
Daan	25	2.53	M	Rock	Cats	Sushi	Autumn	Hiking
Isa	28	4.18	V	Classical	Dogs	Pizza	Autumn	Nature retreat
Lucas	29	5.30	M	Pop	Birds	Pizza	Winter	Nature retreat
Thijs	30	5.85	M	K-Pop	Turtles	Burgers	Summer	Hiking
Nina	31	6.30	V	Rock	Turtles	Pizza	Spring	Citytrip
Naomi	32	4.07	V	Pop	No pets	Sushi	Summer	Beach
Laura	33	2.72	V	Hip-Hop	No pets	Pizza	Autumn	Nature retreat
Ruben	34	4.23	M	Rock	Dogs	Burgers	Winter	Citytrip
Tess	35	4.23	V	Hip-Hop	Dogs	Pasta	Summer	Hiking
Eline	35	5.43	V	Indie	Cats	Pizza	Spring	Nature retreat
Stijn	35	3.48	M	Rock	Cats	Pasta	Autumn	Citytrip
Mila	36	1.82	V	Hip-Hop	Dogs	Pizza	Spring	Hiking
Emma	37	3.77	V	Pop	Dogs	Sushi	Winter	Citytrip
Femke	39	4.62	V	K-Pop	Turtles	Pizza	Winter	Beach
Julia	39	3.69	V	K-Pop	No pets	Burgers	Summer	Hiking
Fleur	40	6.18	V	Hip-Hop	Dogs	Pizza	Summer	Hiking
Iris	40	5.14	V	Pop	Dogs	Pasta	Summer	Beach
Lars	40	4.47	M	Pop	Fish	Pasta	Autumn	Hiking
Sanne	40	2.67	V	Pop	Fish	Pasta	Autumn	Nature retreat
Lotte	41	4.98	V	Hip-Hop	Cats	Salad	Summer	Citytrip
Sophie	45	1.03	V	Pop	Dogs	Burgers	Summer	Nature retreat
Tom	48	3.42	M	House	Cats	Salad	Summer	Beach
Sander	51	4.57	M	Hip-Hop	Dogs	Salad	Summer	Nature retreat
Eva	52	3.85	V	Pop	Fish	Pasta	Summer	Hiking
Lieke	53	1.00	V	Indie	Dogs	Pizza	Summer	Citytrip
Jelle	54	4.50	M	Indie	Dogs	Pizza	Summer	Nature retreat
Tim	55	5.80	M	Indie	Birds	Chocolate	Summer	Beach
Jesse	55	6.24	M	Pop	Fish	Burgers	Spring	Hiking
Bram	55	6.80	M	Rock	Birds	Chocolate	Spring	Nature retreat
Amber	60	6.65	V	K-Pop	Cats	Pasta	Spring	Nature retreat

D Scratch

One of the approaches that turned out to be less suitable was an implementation of NN in Scratch. The aim of Scratch is to stimulate computational thinking by providing a programming environment that has a lower barrier to entry than conventional programming. This environment consists of common programming primitives like for loops, if statements and variables. In scratch these primitives are represented by blocks that operate on sprites. By combining these blocks together the users can create programs, like their own games.

Due to its beginner friendly nature, Scratch can also be used in a teaching environment where the students have little to no programming experience. Every scratch program published on their website can be opened in the editor. In this editor, users are able to explore the inner workings of the program.

Research has shown that Scratch can be an effective learning environment to teach the basics of programming. Next to that, it has also shown that after working with Scratch, students show a more positive attitude towards related fields, for example mathematics (Moreno-León & Robles, 2016).

In this teaching activity students would interact with the NN algorithm in Scratch. The Scratch environment allows them to explore the blocks that were used to create the algorithm. The expectation is that this will aid in the learning process.

The user interface built in Scratch is shown in figure D.1. It can be seen that the center-most point takes on the colour of the closest neighbour, and that this colour changes when the point is moved.

This demonstration shows that it's possible to implement a version of the NN algorithm in Scratch. However, the implementation requires multiple workarounds to function correctly. It has to continually check for changes in the position of the points to determine the distances between them. It was decided that this approach differed too much from the theoretical explanation of NN, which would be confusing and counterproductive for the students. The demo is shared on the Scratch site: `https://scratch.mit.edu/projects/955596918`

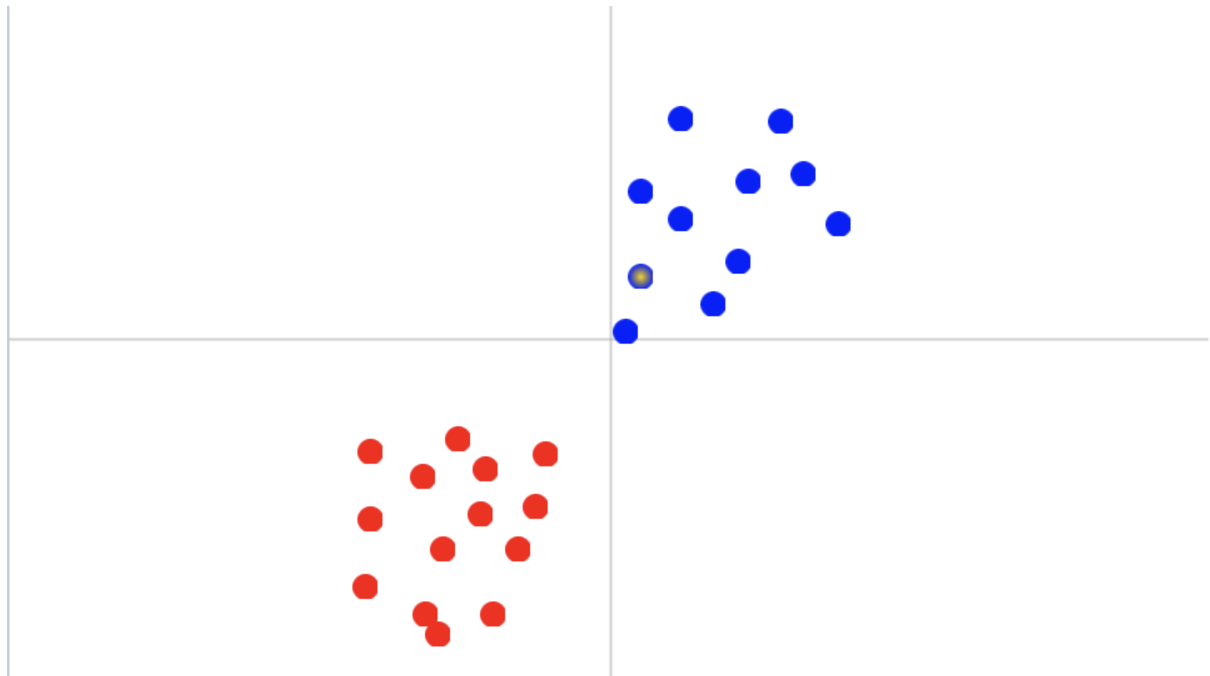

Figure D.1: The UI in Scratch