Table of Contents

Background

Machine learning (ML) is a method of using a computer to inspect features from example sets and then form what is called classifiers from these features. There are are variety of machine learning (ML) algorithms which have developed over the past few decades. In 1983 Machine Learning: The AI Approach was published which is really the foundation for today's ML algorithms. Rather than a book it was a volume of compiled research papers discussing the diverse ML algorithms and approaches to constructing them. [1]

The problem which is being aimed to be solved in this project is a moving slider puzzle. What we want to optimize is the amount of blocks moved to reach a final desired placement of blocks. Figure 1 shows a possible solution using an algorithm known as hill climbing search. This algorithm examines nodes near the current state and then takes the path which has the lowest cost.

What needs to be mentioned about the hill climbing search algorithm is that its does not always lead to the optimal/best path[1]. This algorithm hence has a heuristic approach to solving a problem.

In Figure 1 the cost is the sum of the distances each block needs to take to get into the final positions from the current state positions depicted in the final state. Figures 2 and 3 shows how to calculate a distance in the way that was just mentioned.

Now there are a few machine learning methods to implement the hill climbing search, among them being the four groups of unsupervised learning, supervised learning, deep learning and reinforcement learning. Supervised learning in this case is not too useful as it uses predefined/known labels which in this case would mean a solution that requires something like a data set of completed steps from different start states to the final states.
There are 9^9 = 387.42 * 10E6 possible states for a nine by nine board and generally N^N which quickly grows large and hence brute force computation will be slow. So a more intelligent approach has to be taken to solve this problem.

1 Note an algorithm that always tries to find only the most optimal path is known as a greedy algorithm

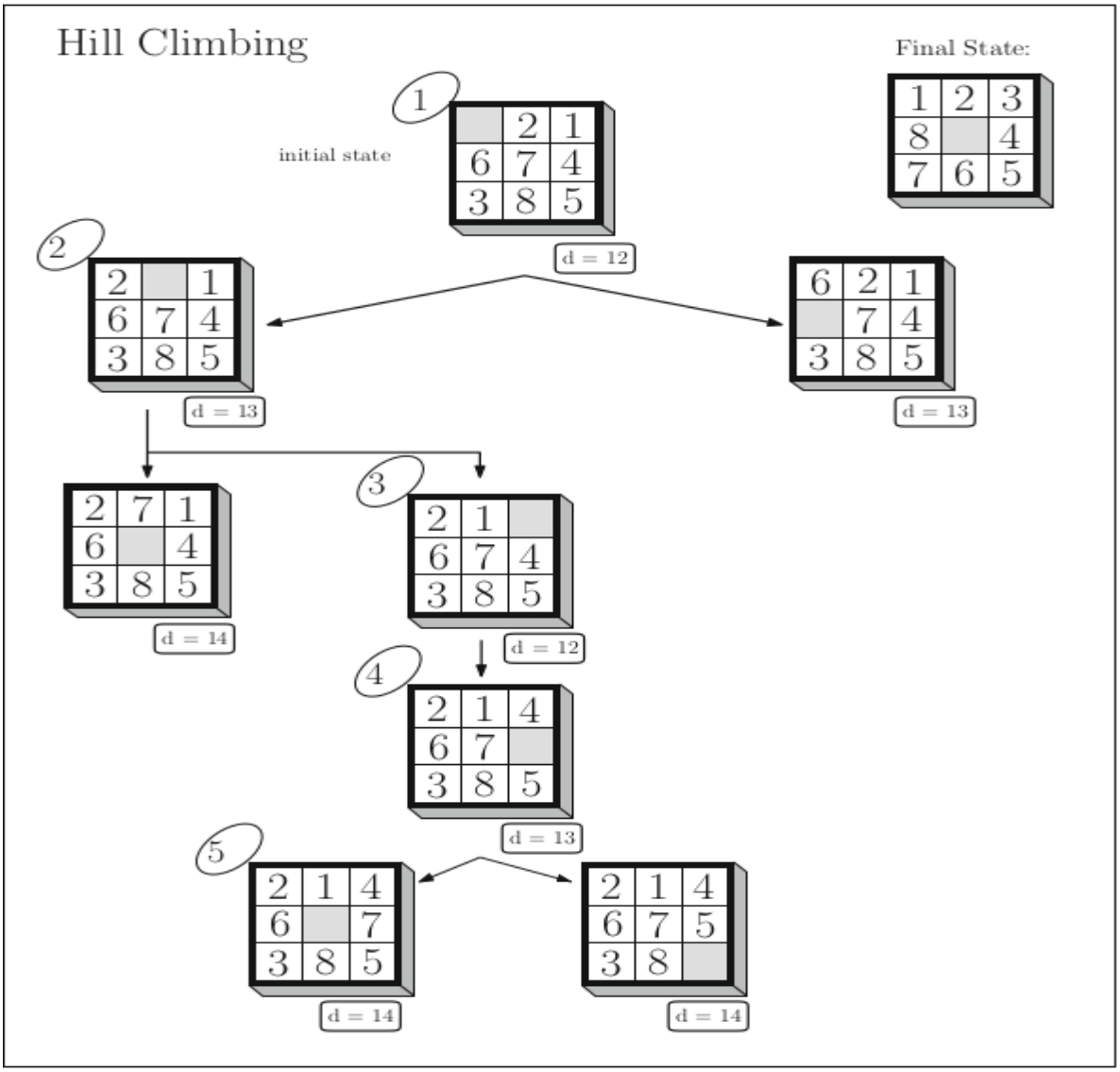

Figure 1 "Hill climbing. Circled integers indicate the order in which the search states are visited.

d is a state's distance from the final state as calculated by the given evaluation function. Ties are broken randomly" [1]

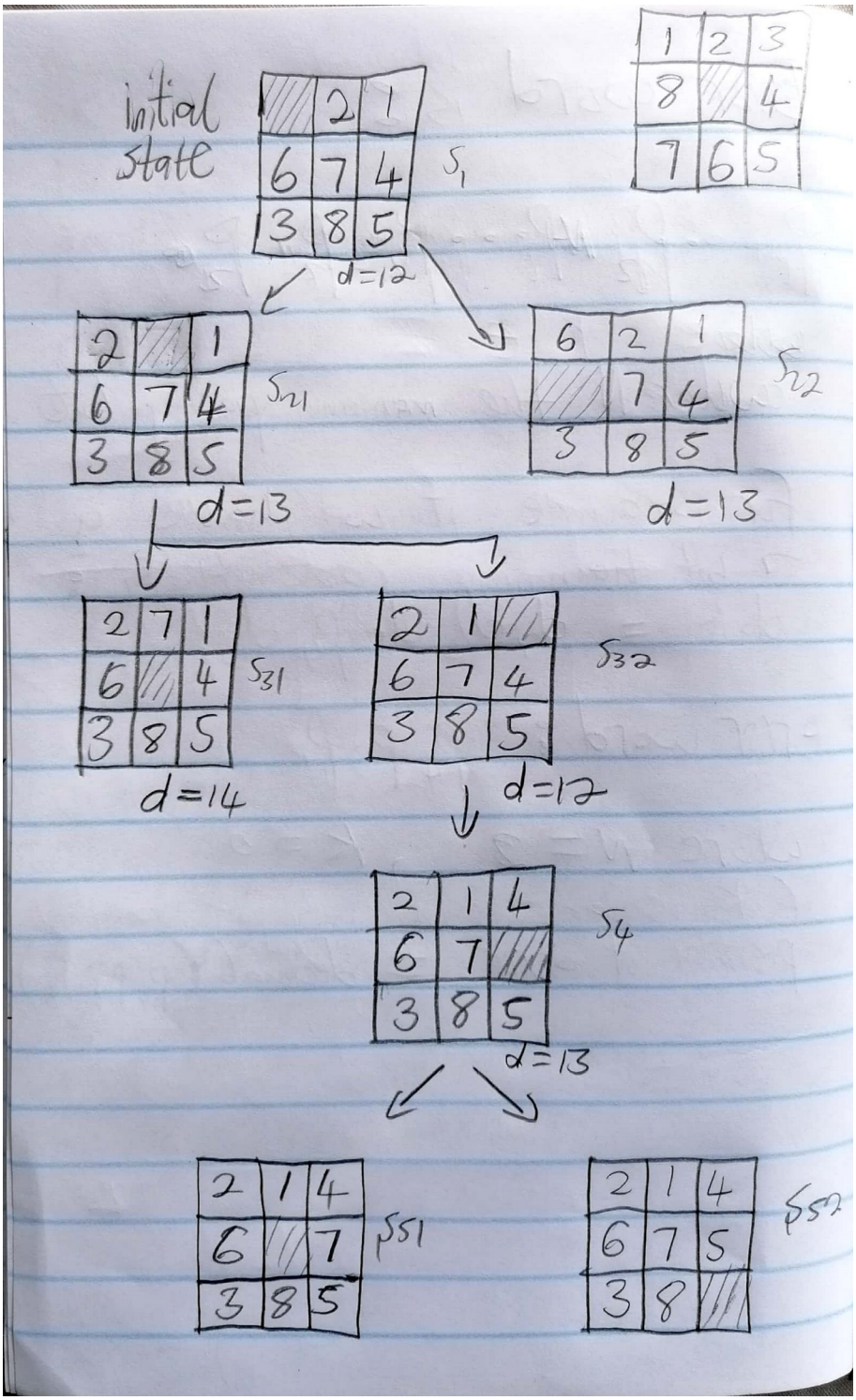

Figure 2 Defining state variable names next to each state to make referencing in the text easier

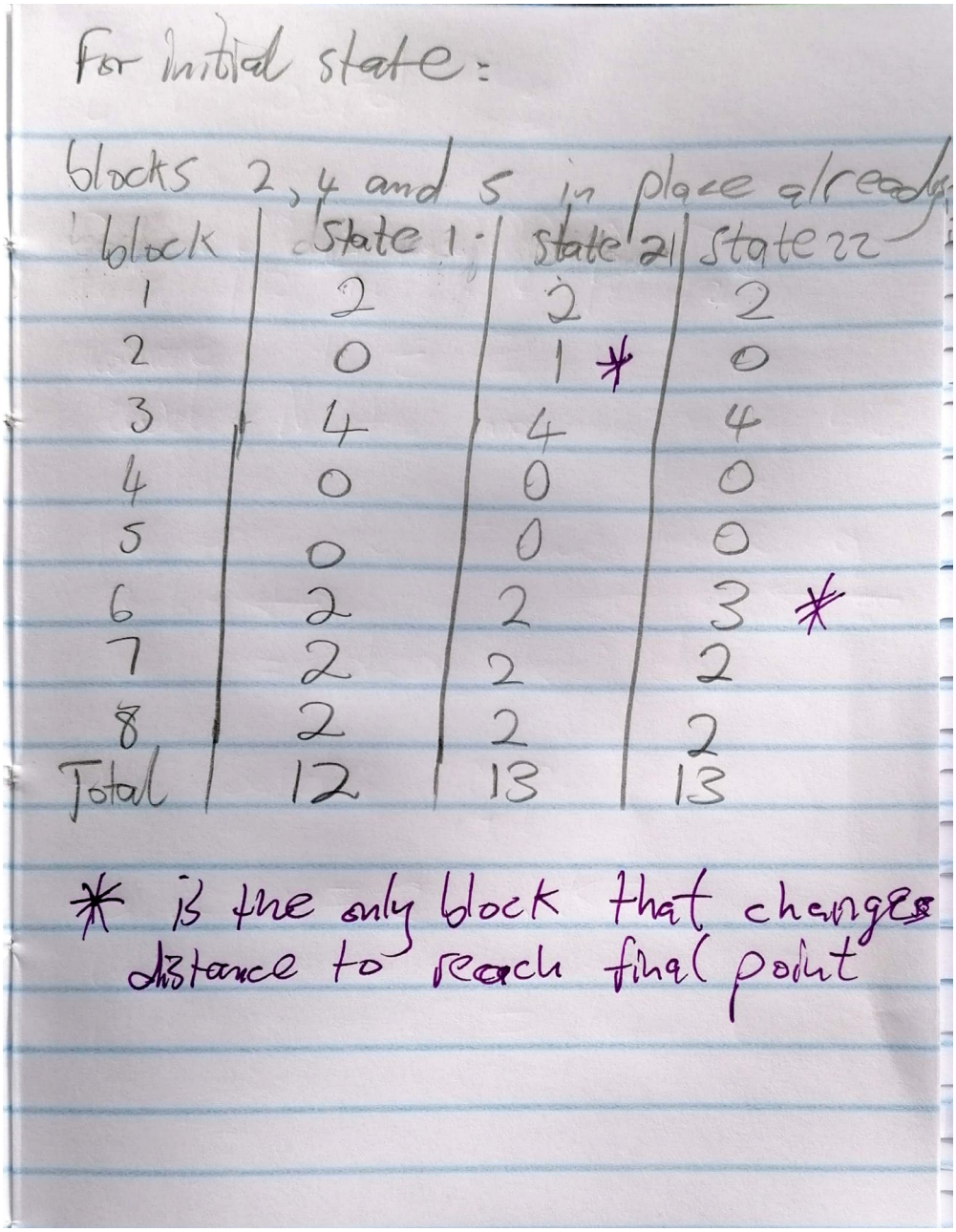

Figure 3 Calculations for the distances in states s1, s21, s22 using state names as defined in Figure 2

Comparison of learning methods

Now there are a few methods to implement the hill climbing search, among them being the four groups of unsupervised learning, supervised learning and reinforcement learning.
Supervised learning in this case is not too useful as it uses predefined/known labels which in this case would mean a solution method that requires something like a data set of completed steps from different start states to the final states. This would require an overly complex implementation. A general algorithm implementation of supervised learning is shown in Figure 4.

Unsupervised learning much more applicable than unsupervised learning as it does not require labeled data and might be able to solve this particular slider puzzle problem by simply having a bunch of start states with their corresponding steps to reach the final state as data. But it would still be tedious to implement as the data is sets of states. The process for supervised learning can be seen in Figure 5.

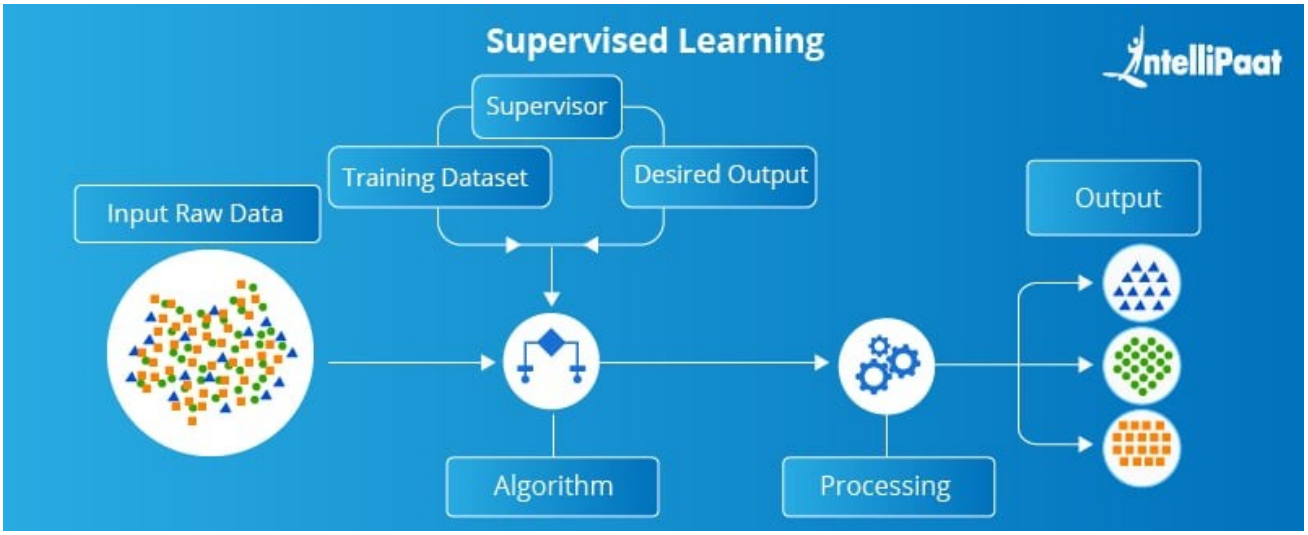

Figure 4 Supervised learning process [2]

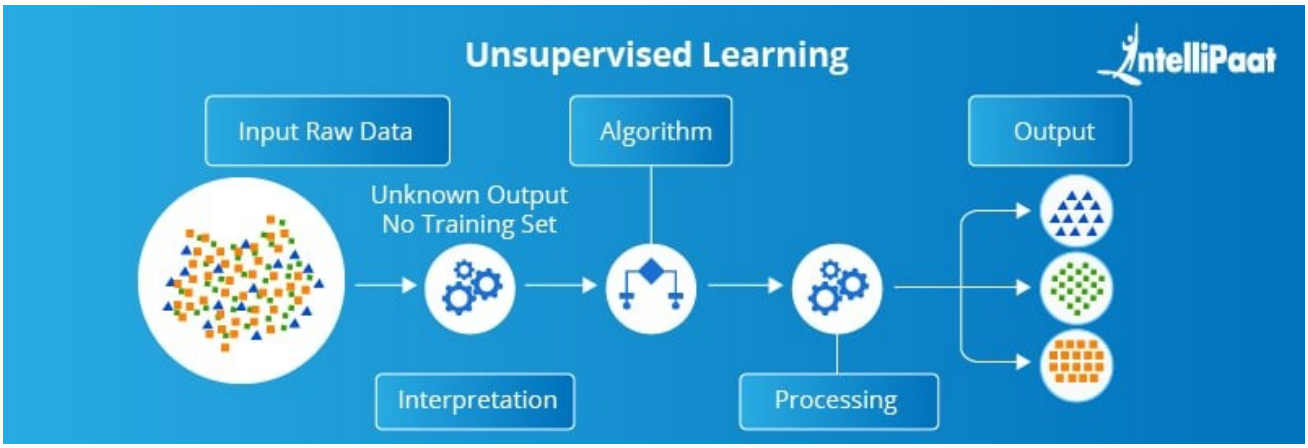

Figure 5 Unsupervised learning process [2]

Lastly there is the reinforcement learning method. Solutions in this case is made by optimizing the path taken from start to finish with regards to some reward for taking a correct step in the correct direction and punishment for incorrect steps. No data is needed before hand for this algorithm and its learns in real time similar to unsupervised learning. Because we do not need data before hand, and the algorithm works real time it is the most applicable and efficient for calculating the best path for the sliding solver puzzle from start to finish.
Hence reinforcement learning was chosen for this projects method of solving.

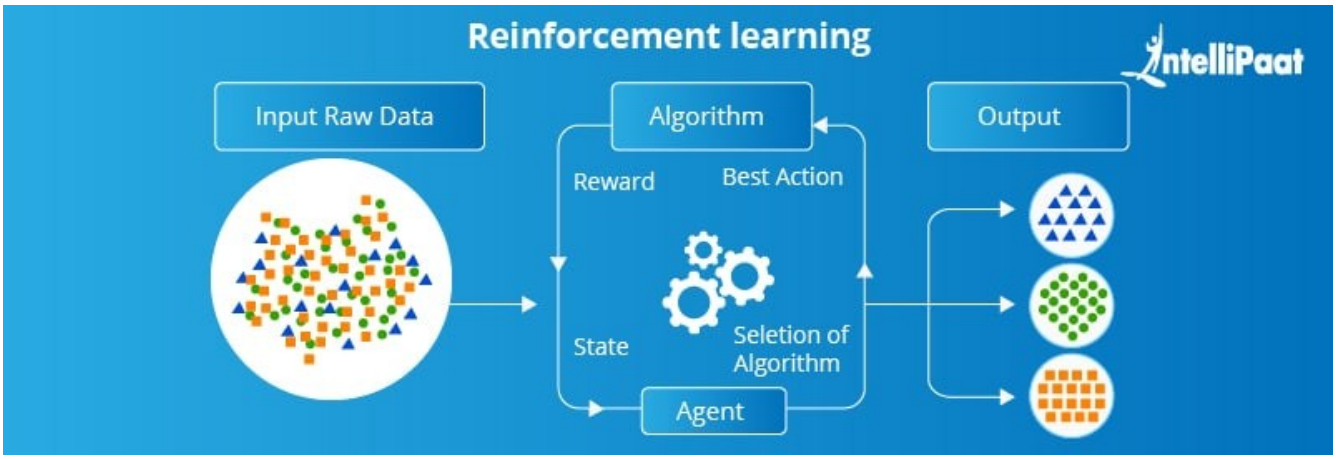

Figure 6 Reinforcement learning process [2]

Figure 7 shows a graph where four different problems and methods are plotted. The approach that will be used for this project will be classic reinforcement learning. The deep learning reinforcement method might be more efficient at solving this problem but for simplicity we will stick to the classic method first which can then always later be morphed and modified to fir the deep learning method.

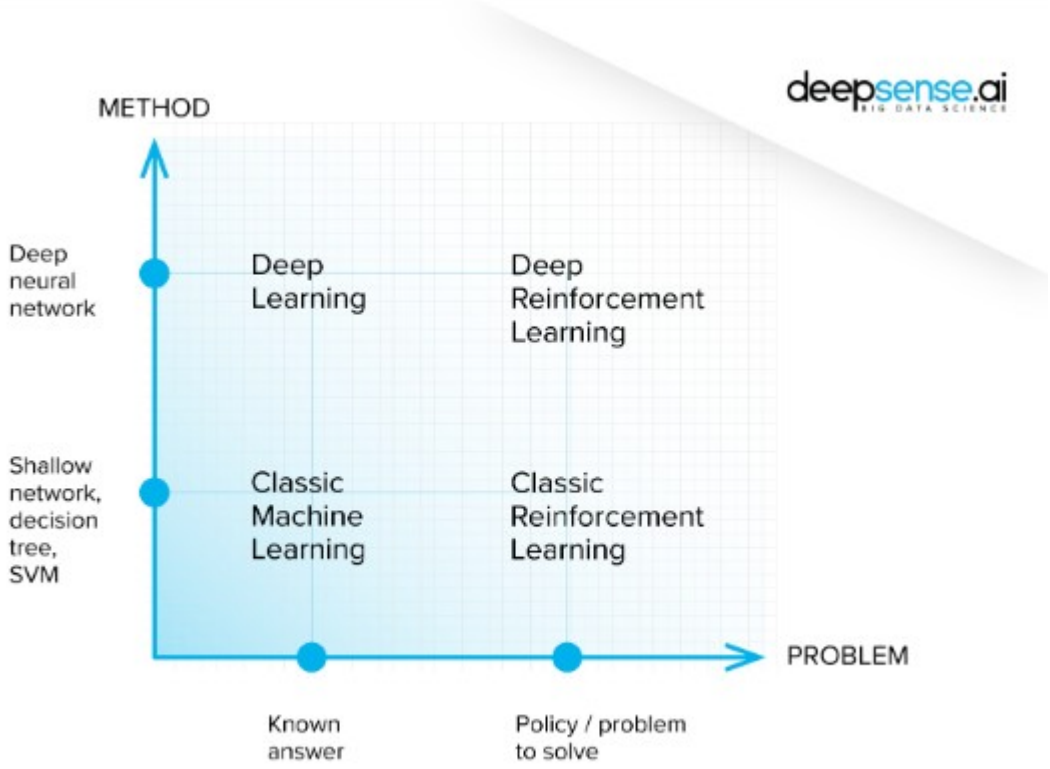

Figure 7 Different learning types in a graph form [3]

Conclusion

In this short report a background was given on machine learning, then a comparison was made between learning algorithms. Finally it was concluded after having compared the learning methods that classic reinforcement learning is the optimal method for this project.

References

An Introduction to Machine Learning Second Edition, Miroslav Kubat, Springer International Publishing AG 2015, 2017 [1]

Supervised Learning vs Unsupervised Learning vs Reinforcement Learning, https://intellipaat.com/blog/supervised-learning-vs-unsupervised-learning-vs-reinforcement-learning/,Intelipaat**,** 26th December 2020 [2]

What is reinforcement learning? The complete guide, by Błażej Osiński and Konrad Budek, https://deepsense.ai/what-is-reinforcement-learning-the-complete-guide/, July 5, 2018 [3]