

The Implementation of a Supervised Learning Algorithm in an AI Car Controller

Mathew Wade Alex Taylor
Mathew.W.A.Taylor@hotmail.com

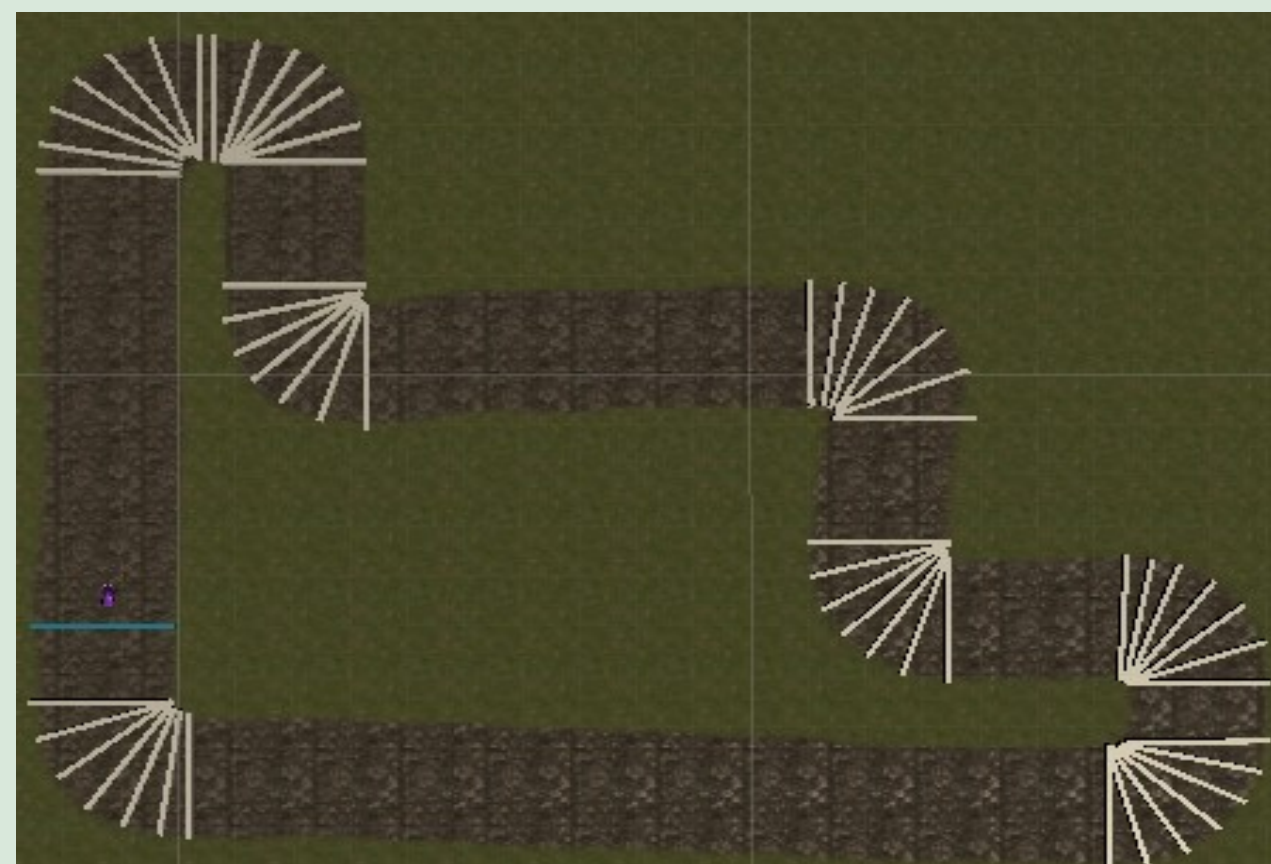
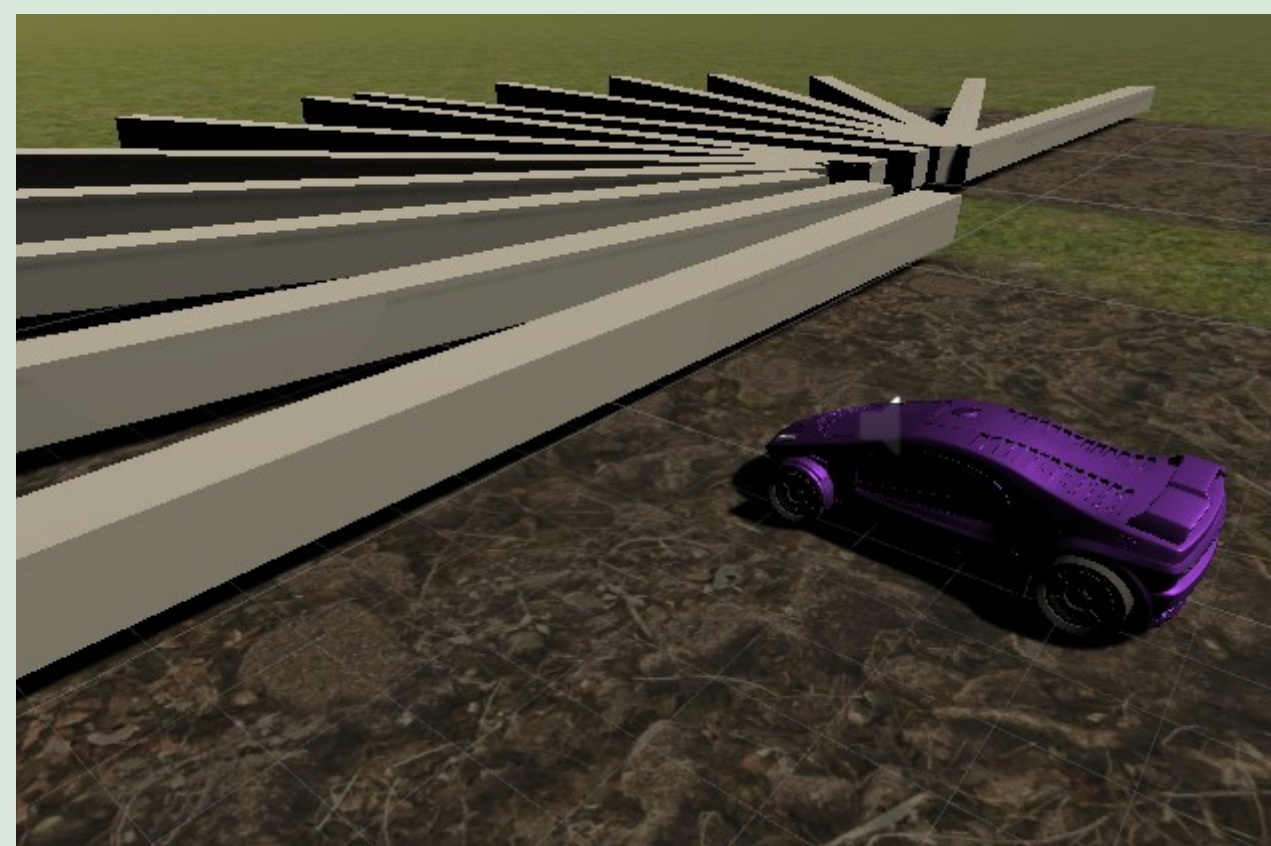
Conclusion

The chosen algorithm is an excellent choice for controlling an AI car

The user is able to drive around a race track in a 3D environment and the correct data is logged to text files

Naïve Bayes calculations were found to be correct, meaning that the AI car's movements were also correct

The AI car is capable of successfully driving around the race track created in Unity as expected



Introduction

Artificial Intelligence is used in a wide variety of ways in games.

The focus of this project is on artificial intelligence in racing games and specifically the development of an AI car that makes use of a supervised learning algorithm to drive around a track in a 3D environment by using recorded data from the user

Aim

To develop, train and evaluate a supervised learning algorithm for use in controlling an AI car in a 3D environment.

Objectives

1. Complete a literature review relating to supervised learning algorithms and relevant information regarding AI cars.
2. Build a 3D race track with relatively realistic car physics and a user controlled car.
3. Log relevant data from the car as the user drives around the track.
4. Implement a supervised learning algorithm into an AI car controller.
5. Evaluate the supervised learning algorithm and improve upon it if needed.

Naïve Bayes Classifier

The chosen supervised learning algorithm is Naïve Bayes Classifier. This method assumes that all variables have independence, meaning that they are unaffected by one another.

- Reads the recorded data and stores it into lists
- Calculates and stores the probabilities of each element
- Uses the Naïve Bayes equation twice using the logarithms of the probabilities relevant to the current action for doing it and not doing it
- The two values calculated are finally compared and the larger value determines the outcome of the action

$$P(A|B_1, B_2, \dots, B_n) = \frac{P(A)P(B_1|A)P(B_2|A) \dots P(B_n|A)}{\sum_{i=1}^m P(B_1, B_2, \dots, B_n|A_i)P(A_i)}$$

Labels in diagram: Posterior Probability (points to the left side of the equation), Prior Probability (points to P(A)), Probability of Predictor Given Class (points to P(B₁|A), P(B₂|A), ..., P(B_n|A)), Prior Probability of Predictor (points to the denominator).

Naïve Bayes Equation after the denominator is replaced with equivalent summation[1]

Uses of Naïve Bayes

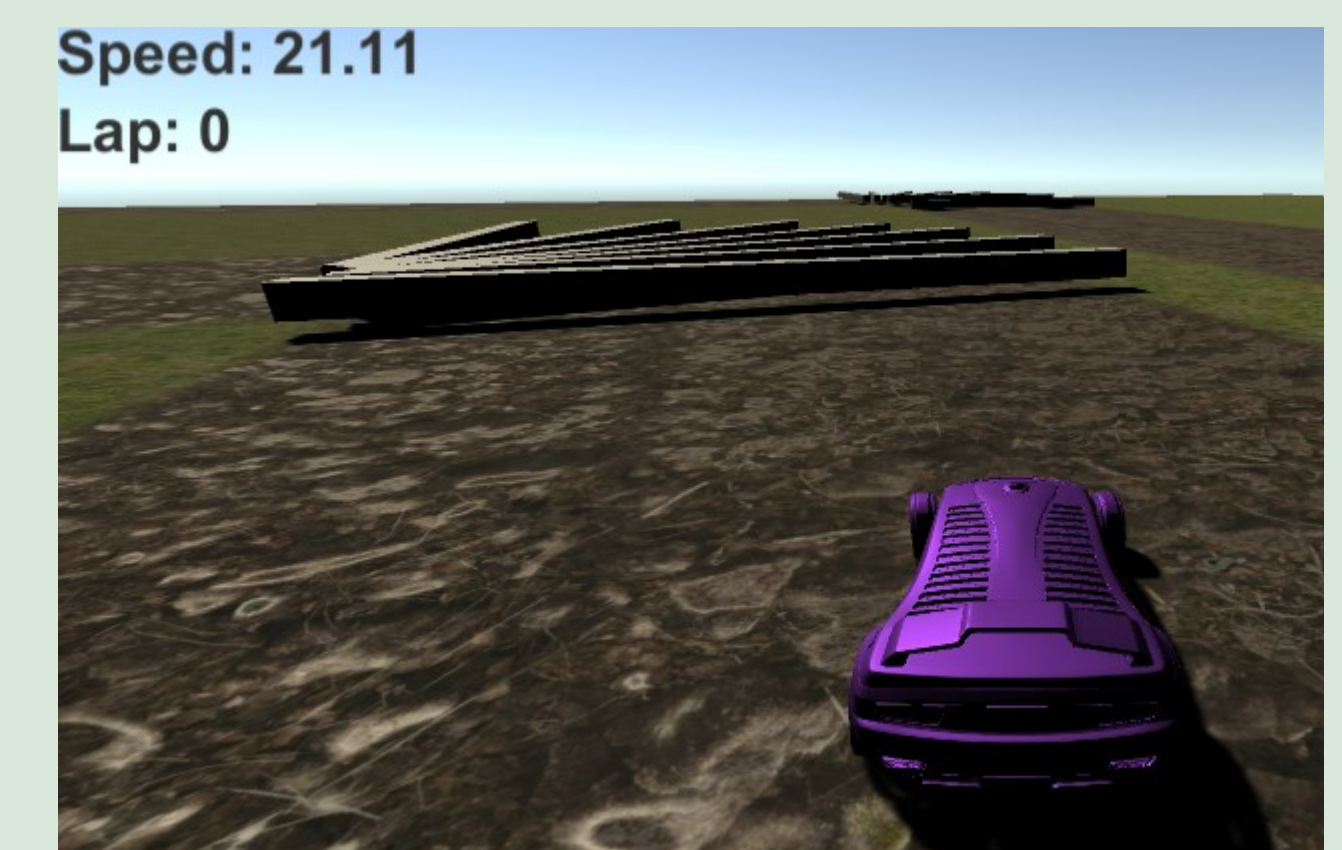
- Braking
- Turning
- Turning Amount

Results

Naïve Bayes results were found to be correct through comparisons made between the Unity debug console and a separate excel spreadsheet doing the same calculations

The AI car brakes and turns as expected allowing it to traverse the whole race track

The AI car driving around the track was found to not be affected by it's starting position and rotation as long as it came in contact with the first waypoint



Future Work

- Record more data
- Improve upon the AI driving along straights
- User vs AI racing
- Implement object avoidance

References

- [1]- D. Berrar, "Bayes' Theorem and Naive Bayes Classifier," in *Reference Module in Life Sciences*, 2018.