

An Implementation of a supervised learning algorithm for AI car control in video games

Matthew Dickinson
BSc Computing (Games Development)

Email: 1708355@student.uwtsd.ac.uk
mattjdickinson96@gmail.com

1. Overview

The project has implemented and evaluated a machine learning algorithm within the context of an AI car controller component within the Unity game engine.

2. Aim & Objectives

Aim:

"To develop, train and evaluate a supervised learning algorithm that can be used within a game engine to autonomously control a car"

Objectives:

- Create a virtual environment and car.
- Log user input and car physics data.
- Prototype Naïve Bayes using collected data.
- Implement the algorithm within Unity.

3. Naïve Bayes Classifier

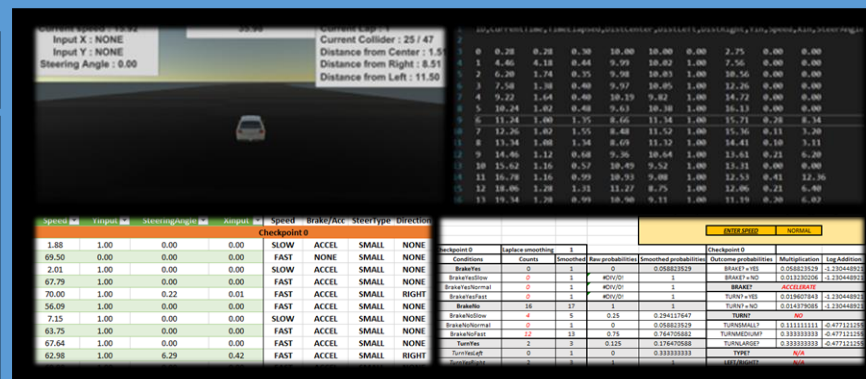
The naïve Bayes classifier is a supervised learning algorithm. It utilises Bayes theorem to estimate the most likely outcome of an event.

$$P(C|x_1, \dots, x_n) = \frac{\prod_{i=1}^n P(x_i|C) \times P(C)}{\prod_{i=1}^n P(x_i)}$$

Within this implementation, the classifier used training data gathered from a user to predict correct actions to take at certain points of the track when at certain speeds.

4. Software Design

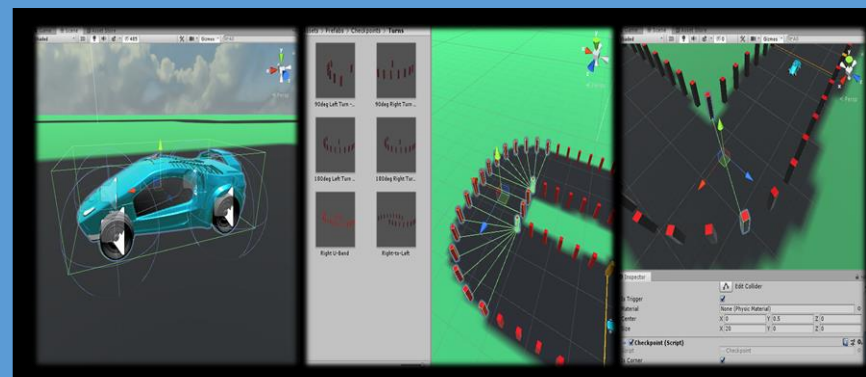
Utilised Feature Driven Development and timeboxing to focus on producing working prototypes that met requirements as needed.



Speed	Input	SteeringAngle	Output	Speed	Brake/Acc	Steering	Direction
1.88	1.00	0.00	0.00	FAST	ACCEL	SMALL	NONE
69.50	0.00	0.00	0.00	FAST	NONE	SMALL	NONE
2.01	1.00	0.00	0.00	FAST	ACCEL	SMALL	NONE
67.79	1.00	0.00	0.00	FAST	ACCEL	SMALL	NONE
70.00	1.00	0.22	0.01	FAST	ACCEL	SMALL	RIGHT
56.09	1.00	0.00	0.00	FAST	ACCEL	SMALL	NONE
7.15	1.00	0.00	0.00	SLOW	ACCEL	SMALL	NONE
63.75	1.00	0.00	0.00	FAST	ACCEL	SMALL	NONE
67.64	1.00	0.00	0.00	FAST	ACCEL	SMALL	NONE
62.98	1.00	6.29	0.42	FAST	ACCEL	SMALL	RIGHT

5. Software Development

Created prototypes of the both the test track and the car. Used collision detection at evenly spaced locations along the track to log training data. Then implemented the naïve Bayes classifier within an excel spreadsheet before finally applying the algorithm to an AI car controller C# script.



6. Conclusion

The naïve Bayes classifier can be used to replicate steering, acceleration and braking behaviours of users, with some adjustment required before entering straight sections to prevent the car leaving the track.

7. Future Work

Some areas where improvements can be made include:

- Smoother adjustment on entering straights
- Higher density of checkpoints to increase detail of training data.
- Increase the size of the training dataset to improve reliability of controller outputs.
- Implementation of other supervised learning algorithms for comparison.

References

- [1] I. Millington, Artificial Intelligence for Games, 2nd ed. CRC Press, 2009.
- [2] H. Zhang, "The Optimality of Naive Bayes," 2004.