Prediction Iteration Method

Method Summary: The Prediction Iteration Method creates an initial training dataset and trains a model with it. It pushes this initial model to an array. It then creates a new dataset and loops through it. For every "no move" it finds, it checks if the previous iteration can classify it. Once it is done looping though, it creates a new model for this data and pushes it to the array. This method loops through new datasets over and over until an iteration is created that has no "no moves". When trying to predict what move to do, it starts with the first iteration and keeps going down the array until one model suggests something besides "no move".

Result Summary: The most interesting thing about this method is the dramatic jump in accuracy from a 6x6 training size of 1,000 to 10,000. KNN preformed *okay* with 1,000 and Decision Tree preformed awfully. However, with a training size of 10,000, both made a jump to 100% accuracy. However, the 10x10 board with a training size of 10,000 looks similar to a 6x6 with a training size of 1,000. Naïve Bayes and Discriminant Analysis could not converge with this method.

Board	Training Size	ML Algorithm	Train Time (s)	Accuracy
6x6	1,000	Decision Tree	12.01	25.8%
6x6	1,000	KNN	17.18	78.5%
10x10	1,000	Decision Tree	37.92	6.4%
10x10	1,000	KNN	28.49	16.2%
6x6	10,000	Decision Tree	127.49	100%
6x6	10,000	KNN	255.15	100%
10x10	10,000	Decision Tree	235.70	23.3%
10x10	10,000	KNN	390.90	70.9%

Prediction Method

Method Summary: Same as the Prediction Iteration Method, but instead of creating an array of models and progressing through that array to determine a move, it just creates a single model.

Result Summary: Much better than the Prediction Iteration Method, but still not acceptable. There were two models with high accuracy, but it is inconsistent. Naïve Bayes and Discriminant Analysis still could not converge with this method.

Board	Training Size	ML Algorithm	Train Time (s)	Accuracy
6x6	1,000	Decision Tree	11.77	21.9%
6x6	1,000	KNN	19.14	51.7%
6x6	10,000	Decision Tree	123.62	98.3%
6x6	10,000	KNN	242.96	99.1%
10x10	10,000	Decision Tree	205.77	18.60%
10x10	10,000	KNN	413.41	43%

Delta Method

Method Summary: The delta method creates a single data set (≥ 1000). When the initial data is generated, the robots that are immediately next to the destination are labeled. After that, the script loops through the data. For every "no move", the script checks if moving the robot up/down/left/right creates a pattern that has the same Δd (distance between robot and destination for row and column) of something that is already labeled. The script keeps looping through until all "no moves" are gone. Then, a single model is created using the dataset with no "no moves".

Result Summary: Overall best results so far. Every ML Algorithm was able to converge, training time was very quick, and accuracy was high and consistent. Take note of Discriminant Analysis—all scored an accuracy of 100%.

Board	Training Size	ML Algorithm	Train Time (s)	Accuracy
6x6	1,000	Naïve Bayes	7.68	66.5%
6x6	1,000	Decision Tree	8.69	46.6%
6x6	1,000	KNN	12.61	57%
6x6	1,000	Discriminant	13.59	100%
6x6	10,000	Naïve Bayes	99.80	75.3%
6x6	10,000	Decision Tree	139.94	53.5%
6x6	10,000	KNN	105.45	46.9%
6x6	10,000	Discriminant	78.01	100%
6x6	100,000	Naïve Bayes	881.68	37.7%
6x6	100,000	Decision Tree	807.14	97.3%
6x6	100,000	KNN	797.56	52.9%
6x6	100,000	Discriminant	968.17	100%
10x10	10,000	Naïve Bayes	506.11	51.9%
10x10	10,000	Decision Tree	581.31	33.4%
10x10	10,000	KNN	514.24	37.3%
10x10	10,000	Discriminant	548.66	100%

Delta Method (w/ Popularity)

Method Summary: Same as the Delta Method, but instead of breaking from the classification loop as soon as another Δd pattern is found, it adds up how many ups, downs, lefts, and rights are suggested and picks the most popular one, the one with the highest count.

Result Summary: This took much longer to train and had little effect. It made the Decision Tree and KNN slightly better, but not enough to be great models. Discriminant stayed at 100%. Further tests were not ran because training took too long for a sample size of 10,000 (> 1 hour). Naïve Bayes did not converge.

Board	Training Size	ML Algorithm	Train Time (s)	Accuracy
6x6	1,000	Decision Tree	56.8	49.3%
6x6	1,000	KNN	56.64	72.4%
6x6	1,000	Discriminant	56.12	100%

Delta Iteration Method

Method Summary: The Delta Iteration Method is a combination of both the Prediction Iteration Method and the Delta Method. A single dataset is created and initial model is trained and pushed to a model array. From there, the method loops through the dataset looking for "no moves" that can be moved to create the same Δd of coordinates that are already labeled. For every loop, a new iteration is pushed to the model array. When trying to predict what move to do, it starts with the first iteration and keeps going down the array until one model suggests something besides "no move".

Result Summary: Once again, Discriminant Analysis did very well. However, the plain Delta Method is quicker and just as accurate. Using the iterations on top of the Delta Method dramatically increased the time it took to train. Tests with training size of 10,000 will be ran but are not included in this report as they took longer than 2 hours to converge. Naïve Bayes could not converge at all.

Board	Training Size	ML Algorithm	Train Time (s)	Accuracy
6x6	1,000	Decision Tree	128.39	63.7%
6x6	1,000	KNN	123.18	83.1%
6x6	1,000	Discriminant	137.6	93%