CMSC 436: Artificial Intelligence
Project No. 2
Due Wednesday, Oct. 8, 2024 at noon

Student certification:
Team member 1:
Print Name: *Steven Acosta* *Date: 10/02/2025*
I have contributed by doing the following: Coding and documentation

Signed: Steven Acosta

Team member 2:
Print Name: *Sunay Dharamsi* *Date: 10/1/2025*
I have contributed by doing the following: Documentation

Signed:

Team member 3:
Print Name: *Sona James* *Date: 10/04/2025*
I have contributed by doing the following: Working on part of the code and answering the questions for number 3.

Signed: *(you can sign/scan or use e-signature)*

Team member 4:
Print Name: *Tameem El-kaderi* *Date: 10/03/2025*
I have contributed by doing the following: *Confusion Matrices for soft and hard testing*
Signed: ***Tameem El-kaderi***

Pr.2.1 Perceptron-based classifier (10 pts)
In this assignment please use the datasets from Project 1. In the language of your preference (Python, Java, Matlab, C++), implement a perceptron-based classifier that will iterate until the **total error** is:

Dataset	Total Error
A	Epsilon $\varepsilon<10^{-5}$
B	Epsilon $\varepsilon<40$
C	Epsilon $\varepsilon<700$

In addition, you may want to introduce a limit on the maximum number of iterations (let that be *ni*=5,000).
Please normalize the datasets first. Initialize your neuron using random values between (-0.5, 0.5).

Please use:

a) Hard unipolar activation function
b) Soft unipolar activation function

Note: For either activation function, you may need to experiment with different learning rates (alpha). Smaller alpha (especially with hard activation function) may yield smaller errors. When using a soft activation function, in addition to alpha, you can adjust the gain (smaller gain may result in more iterations, but more "stable" convergence).

For activation function in a) perform the following steps for each of the datasets.

1. Choose 75% of the data for training and the rest for testing. Train and test your neuron. Plot the data and decision line for training and testing data (separately). Note and report the final Total Error (TE) of training. Create a confusion matrix for the testing dataset, as well as rates (true positive, false positive, etc), and compare those to the one from Project 1. (2 pts)

2. Choose 25% of the data for training and the rest for testing. Train and test your neuron. Plot the data and decision line for training and testing data (separately). Note and report the final Total Error (TE) of training. Create a confusion matrix for the testing dataset, as well as rates (true positive, false positive, etc), and compare it to the one from Project 1. (2 pts)

3. Compare **training** TE between steps 1 & 2 above. For **testing** datasets, compare accuracy, confusion matrices, and rates between steps 1 & 2 above. Answer the following questions: (1 pt)
    a. Are error rates different, and if so, why?

b. What is the effect of different data sets and the effect of different training/testing distributions of TEs on the accuracy, confusion matrices, and rates (true positive, false positive, etc).?
c. When would you go with step 1 and when with step 2 from above?
d. Comment and discuss.

Repeat steps 1. through 3. **for activation function in b).**

LINK TO ALL PLOTS: plots_p2

a) Hard Unipolar Activation Function

i) Training TE between steps 1 & 2

Dataset	Training%	Testing%	Training TE
A	75	25	0
	25	75	0
B	75	25	50
	25	75	35
C	75	25	1092
	25	75	380

ii) Testing Data comparison of accuracy, confusion matrices, and rates between steps 1 & 2

Dataset	Training%	Testing%	Accuracy	Confusion Matrix				Rates				Project 1 Comparison			
				TP	FP	TN	FN	TPR	TNR	FPR	FNR	TPR	TNR	FPR	FNR
A	75	25	100.00%	500	0	500	0	100.00%	100.00%	0%	0%	100.0%	100.0%	0%	0%
	25	75	99.97%	1500	0	1499	1	100.00%	99.93%	0.07%	0%				
B	75	25	98.40%	494	6	490	10	98.80%	98.00%	2.00%	1.20%	100.0%	74.0%	26.0%	0%
	25	75	98.50%	1489	11	1466	34	99.27%	97.73%	2.27%	0.73%				

Dataset	Training%	Testing%	Accuracy	Confusion Matrix				Rates				Project 1 Comparison			
				TP	FP	TN	FN	TPR	TNR	FPR	FNR	TPR	TNR	FPR	FNR
A	75	25	100.00%	500	0	500	0	100.00%	100.00%	0%	0%	100.0%	100.0%	0%	0%
	25	75	99.97%	1500	0	1499	1	100.00%	99.93%	0.07%	0%				
C	75	25	56.90%	494	6	75	425	98.80%	15.00%	85.00%	1.20%	95.25%	26.65%	73.35%	4.75%
	25	75	54.27%	140	1360	1488	12	9.33%	99.20%	0.80%	90.67%				

a. Are error rates different, and if so, why?
Yes, the error rates are different across datasets. This is mainly because the amount of overlapping data points increases from dataset A to dataset B, then to dataset C. Since dataset A does not have many overlapping points, it becomes easier to classify those points. Whereas dataset B, which has some overlapping points, has a lower accuracy. Lastly, for dataset C, which has the most overlapping points, it makes it harder for the perceptron to classify those points, leading to a much lower accuracy.

b. What is the effect of different data sets and the effect of different training/testing distributions of TEs on the accuracy, confusion matrices, and rates (true positive, false positive, etc).?
As the number of overlapping points increases, both the training and the testing errors also increase. The TPR and TNR decrease depending on the complexity of the data points and how much they overlap. We can observe the effect based on the accuracy percentage. For dataset A, the accuracy was around 100%, for dataset B, it was around 98%, and for dataset C, it was around 55%.

The effect of the 75-25% and 25-75% split was that for step 1, the model was trained with more data and was able to perform better, leading to higher accuracy, especially for the complex dataset C, which achieved a 2% improvement. In dataset A, both the TPR and TNR are approximately 100%, which means the data was easy for the perceptron to separate. In dataset B, both rates dropped slightly, indicating a few minor misclassifications, but the perceptron still performed well in classifying the points. In dataset C, the TNR and TPR rates dropped significantly, indicating that the perceptron had difficulty in classifying those points. In step 1, it misclassifies the small cars (low TNR), and in step 2, it misclassifies the big cars (low TPR).

c. When would you go with step 1 and when with step 2 from above?
Step 1 could be used if we wanted more accurate results, especially when the dataset is noisy or contains a large number of overlapping points. This helps the perceptron learn better because 75% of the data is used for training. Step 2 could be used if we wanted to know how our perceptron would work on a large, unseen dataset, especially if the dataset has less complex or overlapping data.

d. Comment and discuss.

Overall, the hard unipolar activation function performs well on data that has the least overlapping points or data that is linearly separable. It performs poorly on data that has significantly more overlapping data or data that is non-linearly separable. In Step 1, the increased training data helped the perceptron classify data points more accurately (because it had more examples to learn from), leading to a lower error rate and higher accuracy.

b) Soft unipolar activation function

i) Training TE between steps 1 & 2

Dataset	Training%	Testing%	Training TE
A	75	25	0.016010
	25	75	0.015359
B	75	25	39.858527
	25	75	38.613924
C	75	25	645.133350
	25	75	245.915288

ii) Testing Data comparison of accuracy, confusion matrices, and rates between steps 1 & 2

Dataset	Training %	Testing %	Accuracy	Confusion Matrix				Rates				Project 1 Comparison			
				TP	FP	TN	FN	TPR	TNR	FPR	FNR	TPR	TNR	FPR	FNR
A	75	25	100.00%	500	0	500	0	100.00%	100.00%	0%	0%	100.0%	100.0%	0%	0%
	25	75	100.00%	1500	0	1500	0	100.00%	100.00%	0%	0%				
B	75	25	98.90%	493	7	496	4	98.60%	99.20%	0.80%	1.40%	100.0%	74.0%	26.0%	0%
	25	75	98.73%	1484	16	1478	22	98.93%	98.53%	1.47%	1.07%				
C	75	25	66.10%	468	32	193	307	93.60%	38.60%	61.40%	6.40%	95.25%	26.65%	73.35%	4.75%
	25	75	67.07%	1301	199	711	789	86.73%	47.40%	52.60%	13.27%				

a. Are error rates different, and if so, why?
Yes, the error rates are different across datasets. This is mainly because of the amount of overlapping points across datasets. Dataset A has the lowest error rate because it has the least amount of overlapping points, so the perceptron was able to classify these points more accurately. Dataset B has a small error rate because it has a moderate amount of overlapping points, so the perceptron is able to classify these points with little difficulty. Dataset C has a higher error rate because it has a larger amount of overlapping points between both classes (small and big cars), so the perceptron struggles with classifying these points accurately.

b. What is the effect of different data sets and the effect of different training/testing distributions of TEs on the accuracy, confusion matrices, and rates (true positive, false positive, etc).?
Since the complexity of data increases across the different datasets, the training TE also increases significantly because it becomes challenging for the model to accurately classify the data points. The accuracy and TNR of dataset C are lower because the perceptron misclassifies most of the big cars as small cars. The confusion matrix for dataset A and dataset B looks much better than that of dataset C, as the number of false positives and false negatives increases. With step 1, where 75% of the data is used to train the model, we can see that for dataset A and dataset B, the accuracy rate is on the higher side. This is because the model has more data to learn from. With step 2, where only 25% of the data is used to train the model, the accuracy for dataset A is still on the higher side, but for dataset B, we can see that it decreased by a bit. Regarding dataset C, both step 1 and step 2, have a poor accuracy rate. Overall, the total error is higher when less training data is used.

c. When would you go with step 1 and when with step 2 from above?
We would go with step 1 in cases where we need a higher accuracy rate, and especially when the dataset contains more complex data. This would be a good option because having more data to learn from would help the model classify the points more accurately. We would use step 2 in cases where we have simple, less complicated data and where the data is linearly separable. Additionally, in cases where we would want to observe how the model will perform with limited training and with a large unseen data.

d. Comment and discuss
Overall, the soft unipolar activation function performs well on data that has the least overlapping points or data that is linearly separable, such as Dataset A and Dataset B. Even though the soft activation creates smoother transitions compared to the hard activation, it is still challenging for the perceptron to accurately classify data points that overlap.

Important: The data sets list the data points for both types of patterns (“small” car and “big” car).

Extra credit question: When given a data set, how would you approach selection of which data points to use for training, and which data points to use for testing? Note that the algorithm may fail if trained on one type of pattern, and tested on another, different type of pattern. (1 pt)

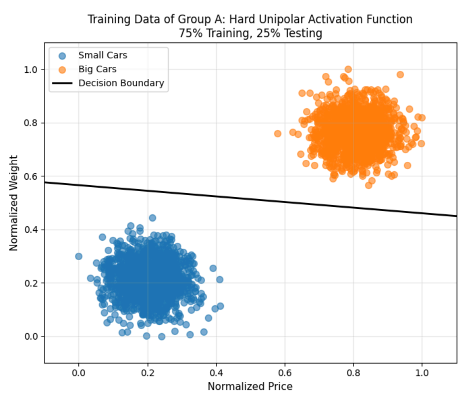
To properly select testing and training data points, stratified random sampling would be best to ensure that both style of cars (small & big) are both being proportionally represented in both sets. This way, the dataset will contain 50% small cars and 50% big cars, both the training and testing sets should maintain a 50-50 ration to prevent any bias towards once class. Datasets should also be randomly shuffled before splitting in order to avoid any systemic bias. This helps prevent failure because if the algorithm is trained on one pattern type and tested on another, then it would never learn the second pattern. With both small and big cars appearing in the training, the perceptron learns how to distinguish between the two. Testing on both types then validates the model's ability to generalize the data not seen from both classes.

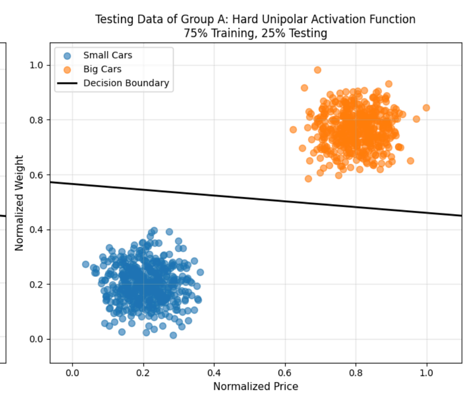
Pr. 2.2 Soft vs. hard activation function (5 pts)

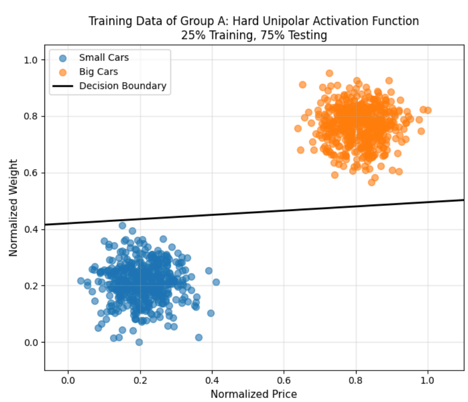
Compare and discuss results obtained with hard unipolar activation, vs. the results obtained with soft unipolar activation function. You should include the plots and provide quantitative comparisons. (3 pts) Comment on each training/testing data split (75% vs 25%, etc), for each data set (A, B, and C). (2 pt)

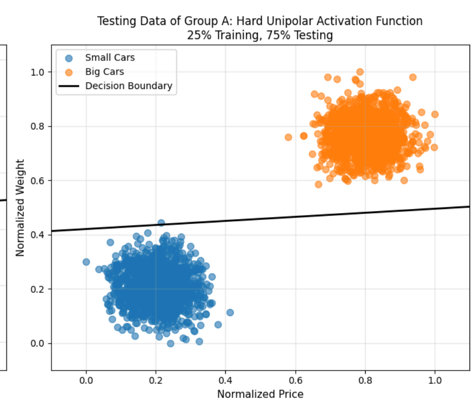
Dataset A

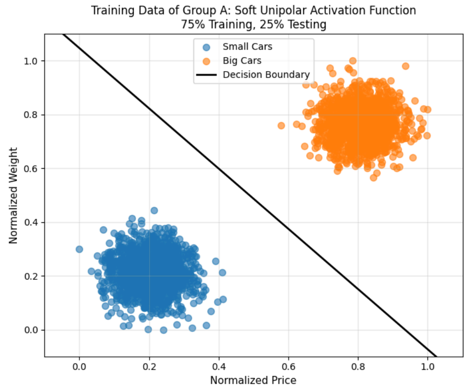
Between hard and soft unipolar activation, the rates of dataset A are pretty uniform between the two different activation types. A majority of the training to testing models of dataset A had an accuracy of 100% between both hard and soft activation; the only exception to this was the 25% - 75% training to testing model for hard activation which had an overall accuracy of 99.7% (one incorrectly predicted negative point). The decision boundary between the testing and training charts in both hard and soft activation as well as 75-25 and 25-75 models is shared for each respective instance of dataset A.

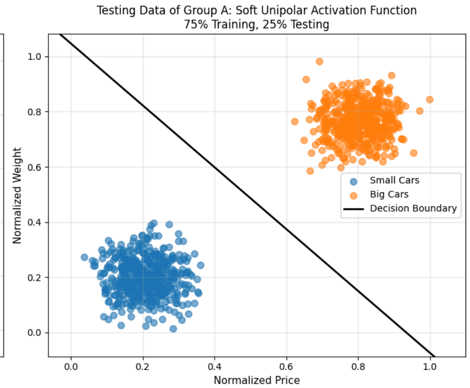


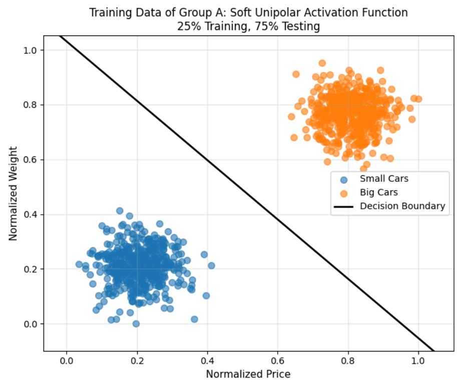


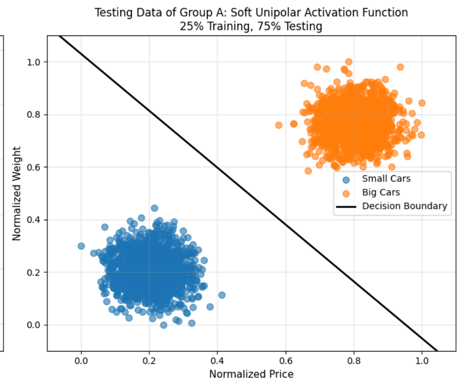






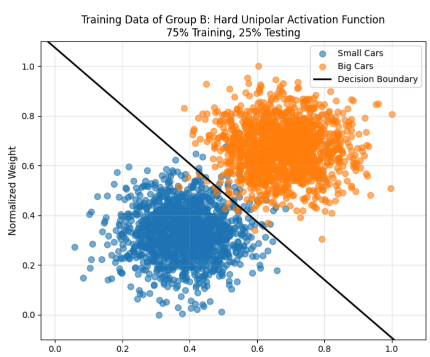


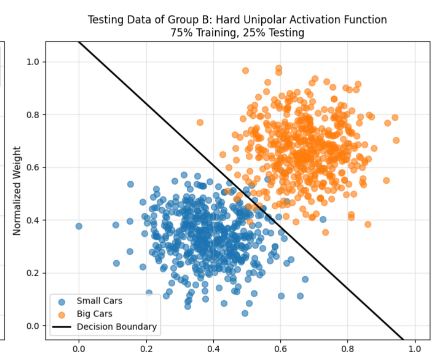


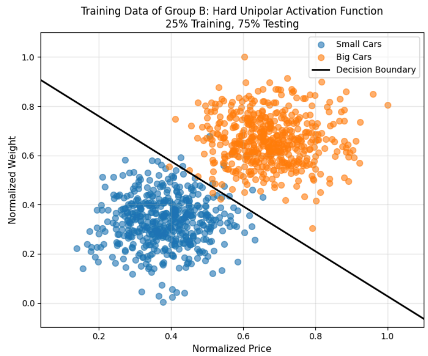


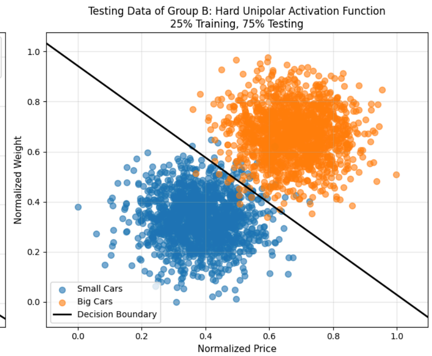
Dataset B

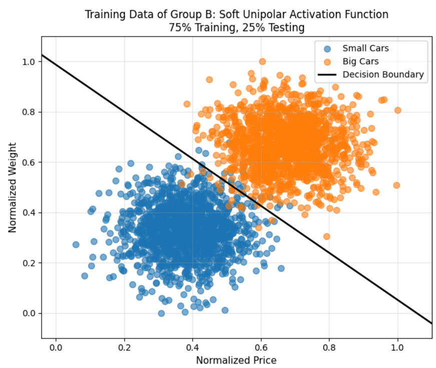
Similar to dataset A, the accuracy for all the 75-25 and 25-75 models for both hard and soft unipolar activation is roughly 99%. As observed by the decision boundaries in the charts below, there exists a distinct split between positive and negative points which indicates a high prediction accuracy for each model. The TPR in the hard activation function for the 75-25 and 25-75 models are 98.80% and 99.27% respectively and the TNR are 98.00% and 97.73% respectively as well. For the soft activation function, the TPR for the 75-25 and 25-75 models are 98.6% and 98.3% respectively and the TNR are 99.2% and 98.53% respectively as well.

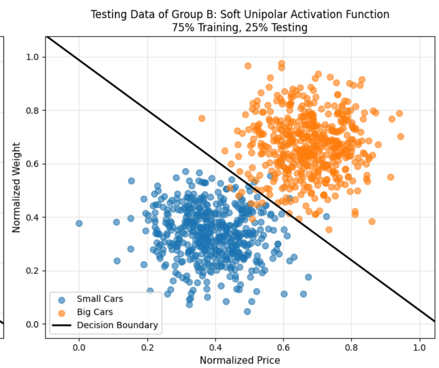


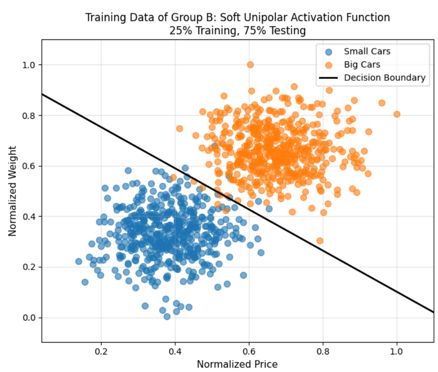


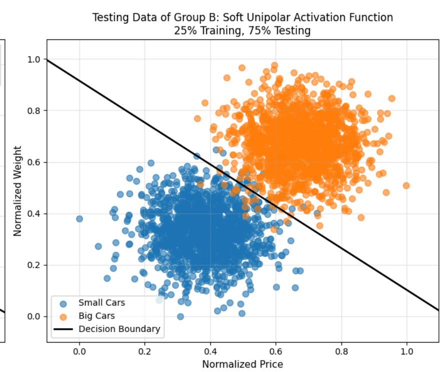






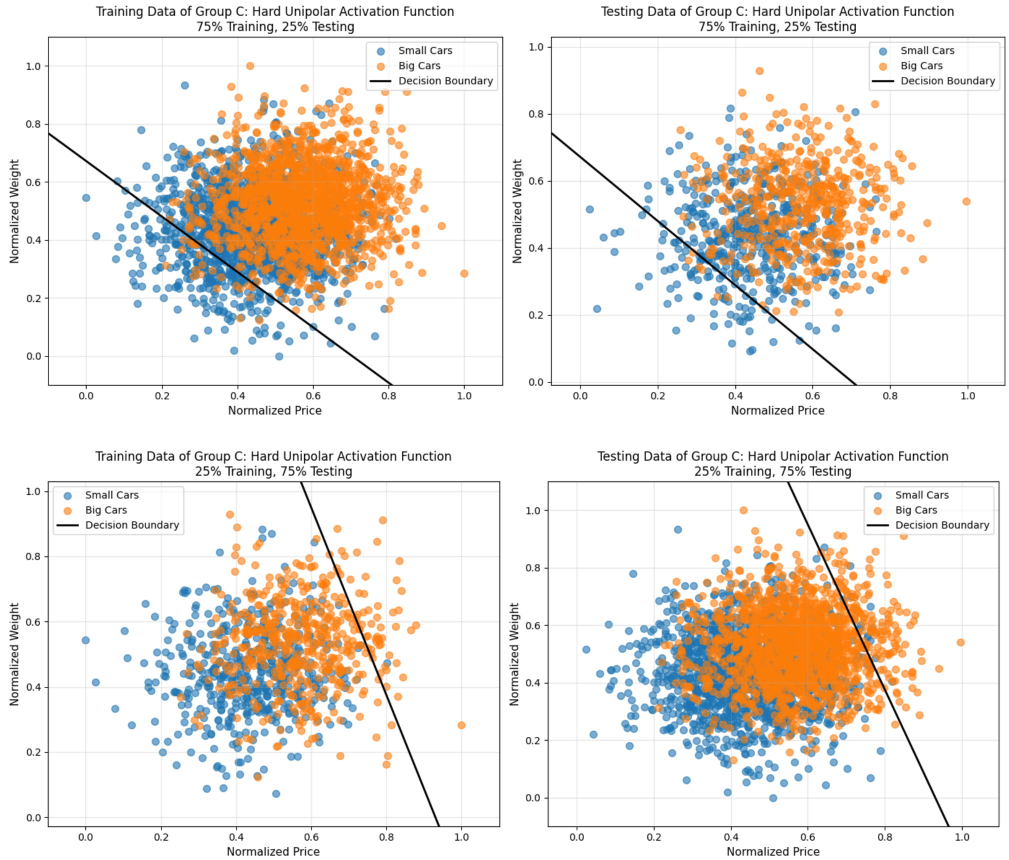




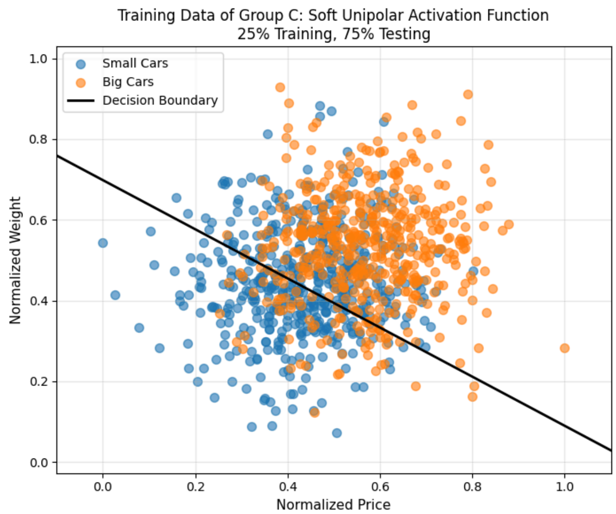


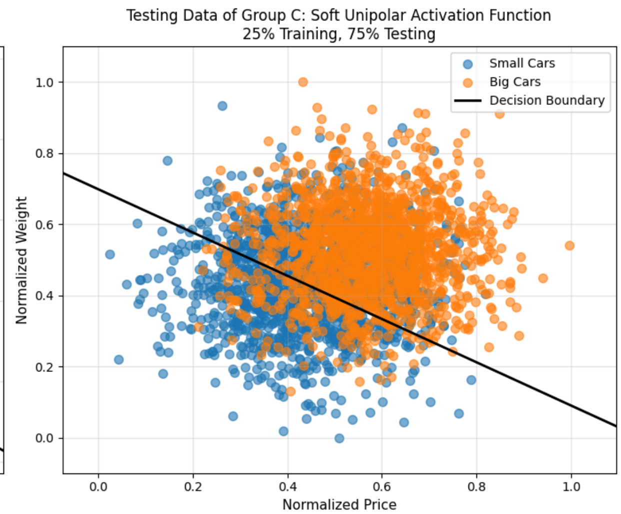
Dataset C

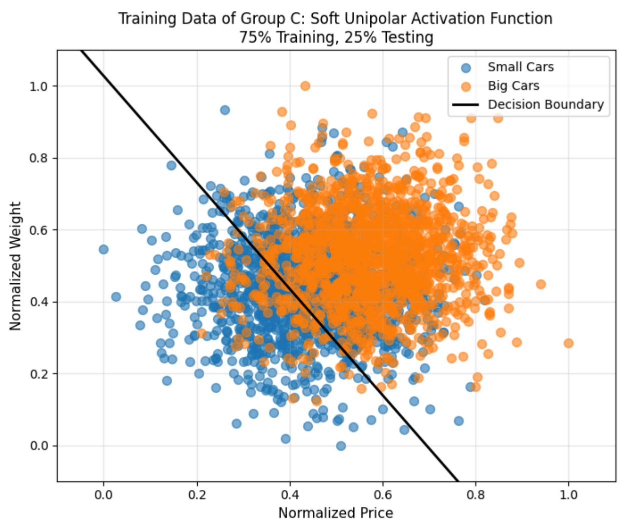
In hard unipolar activation, the training model becomes increasingly inefficient in correctly predicting if a data point is a true positive as the data points begin to overlap more frequently and the training size gets smaller. As seen in Dataset C, in the 25% - 75% model for training and testing respectively, the TPR of the model only amounts to 9.3%, indicating that most of the positive data points (large cars) were incorrectly perceived as negative (small cars). Similarly, the TNR of the 75% -25% model of training to testing comes out to 15.00%, indicating that most of the negative data points (small cars) were incorrectly predicted as positive (large cars). As seen below, the lack of a clear decision boundary between small and big cars is responsible for the large false positive and false negative rates observed.

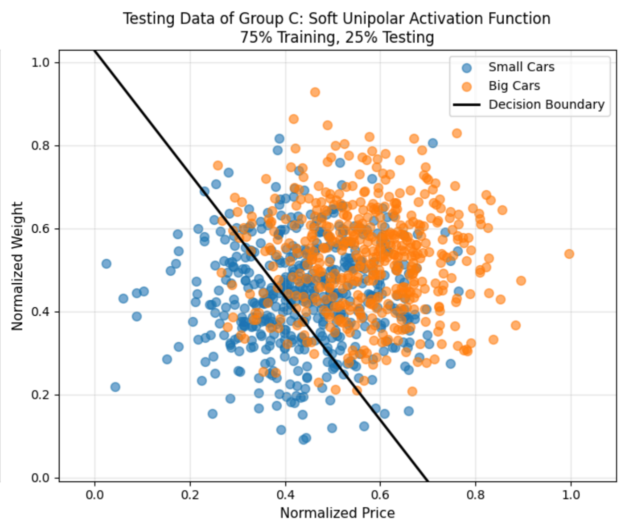


In contrast to hard unipolar activation, soft unipolar activation demonstrated greater consistency between the 75% -25% and 25% - 75% training to testing models in Dataset C when faced with datasets that had no clear line of distinction between positive and negative points. The 75% - 25% model held a TPR of 93.60% while the TPR for the 25% - 75% model was 86.73%. The TNR of both these models was also relatively consistent as well. The TNR in the model with a larger training size had a TNR of 38.60% and the model with a smaller training size had a TNR of 47.40%. In addition to greater consistency, the overall accuracy of the soft unipolar models in dataset C was roughly 10% more accurate than the hard unipolar models for dataset C. As seen below in the comparison between the training and testing models, the decision boundary in dataset C for the soft unipolar models is placed in such a way that a majority of the positive data points are predicted correctly as big cars (above the decision boundary) while the negative data points are split more evenly on either side of the decision boundary.









Note:

1. The code must be user-friendly. The TA must be able to test your code by simply executing it.
2. Project deliverable should be two files:
   a. Written report with all the plots and answers to all of the questions above, in **pdf** format. b. A zip file containing:
      i. Training/testing data sets as specified in Pr. 2.1 steps 1 & 2
      ii. Source code. For example, Python code can be in a Python notebook file (.ipynb) or a Python file (.py).
3. Submit your deliverables (Canvas). Please name files as:
   a. GroupName_Project2.pdf and
   b. GroupName_Project2.zip