Article Summary:

In the example provided:

Two Firms, A and B, choose a price for their product. The prices can be high or low. Depending on which combination of strategies they pick, both companies receive profits according to a defined payoff matrix. We record payoffs depending on which prices each of them pick, using a matrix (e.g. if both pick high prices, one picks high and the other low, etc.).

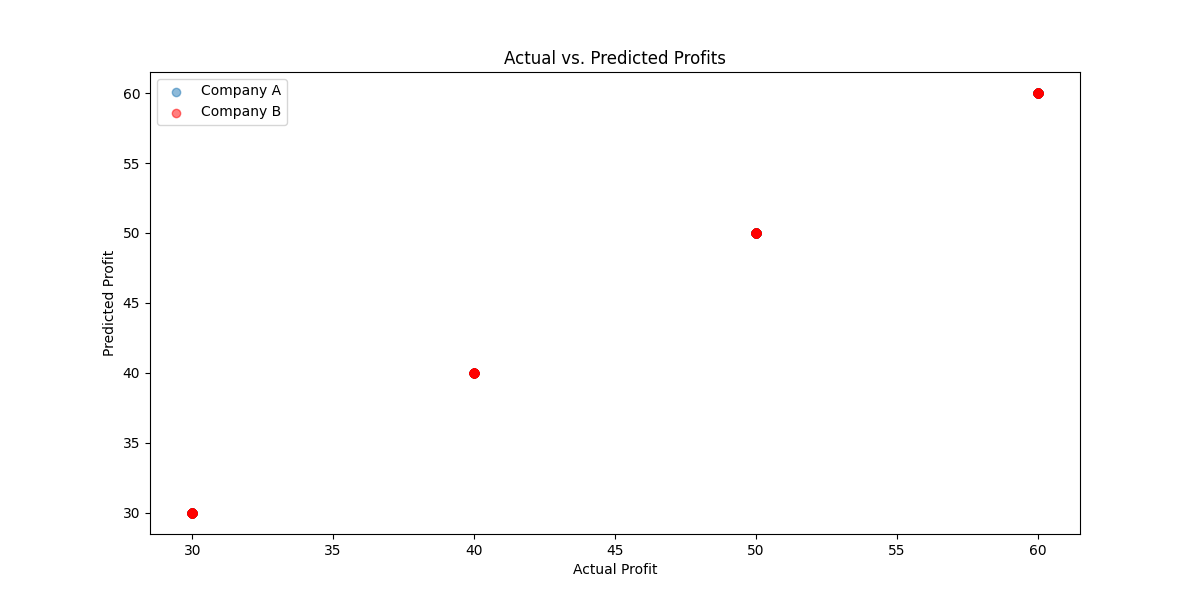
Using a feedforward Neural Network Model, we can learn to predict the companies’ pricings based on their strategies. After training, we can use the model to predict company profits based on a given strategy pair.

Traditional Game theory would do something similar to finding Nash equilibria, and stop there, but using ML helps us find what could possibly be more complicated payoff structures.

This approach is also useful when there are multiple competitors, changing payoff structures, etc., and can be used to build models for more complicated dynamics where Game Theory might not be enough.

The code they provided has been modified and run by me, and it is stored in this repo as pricing-model.py

Below is the result of running the code in pricing-model.py:



Mean Squared Error: 1.3668796650101456e-10

Source:

<https://medium.com/the-modern-scientist/integrating-neural-networks-and-game-theory-enhancing-predictive-analytics-in-strategic-e15258eb4ad1>