**Predicting Sales from Conversations**

Task A.

From given sentiment scores, create a **bi-directional product comparison network**. Use the principles laid out in the article “Product comparison networks” (and discussed in class) to answer this question.

As shown in class, in order to create a directed product comparison network, you need to determine where the arrows should point, and weights for each arrow. First create columns for every possible difference of sentiment scores. E.g., if there are three products, X, Y and Z, the columns should be X-Y, X-Z and Y-Z. Note that it is **not** necessary to also create Y-X, Z-X, etc. The reason is that when the sentiment for Y is larger than that for X, the difference X-Y will be negative, which can be dealt with separately.

Once these columns have been filled up, for each column (e.g., X-Y), do the following:

Take the average X-Y score for all cases where X-Y > 0. If this average score is greater than 0, then there will be an arrow from Y to X (because for these posts, X has a more positive sentiment than Y). The average score is the weight of the arrow. Similarly, for all cases where X-Y < 0, get the average score. If the average score is < 0, then there will also be an arrow from X to Y, since Y has a higher positive sentiment. The average score (ignore the negative sign) is the weight of the arrow from X to Y. Now complete this for ALL difference columns.

**Note: No visual representation of the product comparison network is needed at this stage. It will be required in Task C.**

Task B. Calculate the **weighted PageRank scores** for each car. What is the correlation between the weighted PageRank and sales figures shown below? A sample python script to calculate weighted PageRank scores is posted on Canvas. You have to install the networkx Python library to make this script work. You also need to fill in all the details from your calculations in Task A.

Also run a regression with sales as the output (dependent variable) and the weighted PageRank score as the predictor (independent) variable. How would you use this approach in the real world to predict, say, the sales of a new model of Tesla? Outline the steps (you don’t have to do an actual analysis).

|  |  |
| --- | --- |
| Model | Approximate # sold in the U.S.A. (2012-2013) |
| Audi A6 | 20k |
| Audi A8 | 12k |
| BMW 3-series | 220k |
| BMW 5-series | 60k |
| BMW 7-series | 14k |
| Jaguar XJ | 6.6k |
| Lexus ES | 135k |
| Lexus LS | 30k |
| Lexus RX | 120k |
| Mercedes S-class | 25k |

Task C. For all the links that have a non-zero weight, create a labeled and weighted network using Gephi or NodeXL (works on Windows only). Calculate the PageRank scores ignoring the weights (both tools can calculate these scores). What is the correlation between the PageRank scores and the sales data shown above? Why is this correlation smaller than that between weighted PageRank and sales?

Can you suggest a way to increase the correlation between the PageRank and sales? That is, how can we make the PageRank scores more meaningful? Hint: Make the network sparser by not showing arrows if the weight is not greater than a minimum value. Make the necessary changes (explain what you did), re-calculate pagerank scores and show the new (higher) correlation.