Manufacturers Recommendation System With Natural Language Processing

**Introduction:**

As part of this project we are applying the Natural Language Processing on manufacturing capabilities of each manufacturer and the Input text (**capabilities that user looking for**) and provide the closest results related to the Input text to the user.

To get the data of manufacturers, we followed the web scaring technique by which we collected the data from many online resources such as manufacturing websites, social media etc. After the collection of data we cleaned the data and used the TFIDF vectorizer to convert the input and manufacturer capabilities text into numerical matrix format, and we used the cosine similarity score to get the similarity score between the Input Text and Manufacturer capabilities and we displayed the manufacturers by highest similarity score to the user so that the user will get the suggestions based on the highest similarity score (suggestions that are related closely to the input text).

The detailed steps are mentioned as below.

**1.Collecting the Manufacturers Data:**

We have collected the Manufacturing details through web scraping of multiple online websites and social media etc., The Manufacturer Details contains columns as described below:

Name : Manufacturers name

Address : Manufacturers Address

Manufacturer\_state : State where the manufacturer is present

Number : phone number of Manufacturer

URL : Website Url of Manufacturer

Employees : No of Employees working at Manufacturer

Description : About Manufacturer

Capabilities\_Overall : Capabilities of Manufacturer

**About Capabilities of Manufacturer:**

The Capabilities column in the data table contains the complete list of operations or services that each manufacturer providing,

Below list is the example of capabilities of single manufacturer:

|  |
| --- |
| ['Assembly Machinery', 'Automotive Parts', 'Bending Services', 'Boring Services', 'Broaching Services', 'Custom Machinery', 'COVID-19 Response', 'Cutting Services', 'Cylinders', 'Drilling Services', 'Electronic Components', 'Extrusion Services', 'Fabrication Services', 'Hydraulic Equipment', 'Hobbing Services', 'Job Shop Services', 'Knurling Services', 'Machining', 'Machinery', 'Maintenance and Repair Services', 'Medical Components', 'Milling Services', 'Motors', 'Power Units', 'Plastics', 'Pumps', 'Tapping Services', 'Testing Services', 'Tier 2 Medical Mfg. Supplies, Materials, Components & Services', 'Turning Services', 'Valves', 'Welding Services', 'Winding Services'] |

**2.User Input :**

As Input we will Accept the state of Manufacturer and Manufacturing Capabilities that user is looking for.

**Example:**

Input State = “CA”

Input Capabilities = ["COVID-19 Response"]

**3.Getting the Data based on the Input State**

Based on the state entered as input we will select the manufacturers capabilities from our data which are from the input state and perform the next steps on that data.

**4. Text Normalization:**

In this step we perform the text Normalization on data selected based on state and perform below steps,

* Convert the text into lower case
* Remove unwanted spaces
* Remove unwanted characters in the word
* Tokenize the text

**5. Text Vectorization:**

In this step we performed the text vectorization for both Input text and the capabilities of the manufacturer using the TFIDF Vectorizer.

**TFIDF Vectorizer:**

TF-IDF stands for "Term Frequency -- Inverse Document Frequency". This is a technique to quantify a word in documents.

**TF-IDF = Term Frequency (TF) \* Inverse Document Frequency (IDF)**

**Term Frequency:**

The term frequency of a word in a document is calculated by counting the number of instances a word appears in a document / total no of words in document.

tf(t,d) = count of t in d / number of words in d

t = word in document

d = Document

**Inverse Document Frequency:**

The Inverse Document Frequency of a word is calculated by taking the total number of documents, dividing it by the number of documents that contain a word, and calculating the logarithm.

idf(t) = loge (N/(df ))

t = word in document

N = Total number of documents

df = Number of documents that contain a word

**Example :**

In the below list each element is consider as one document.

['assemblies aircraft parts aluminum cases drilling services engine parts forming services machining manufacturing services rods swaging services shafts stamping services struts steel tube fabrication and bending services titanium tubes',

'aerospace components aircraft parts assemblies automotive parts covid19 response cutting services drilling services electronic components engineering services fabrication services grinding services machining milling services medical components programming services prototyping services turning services tapping services tier 2 medical mfg supplies materials components services tube fabrication and bending services welding services',

'assembly services fabrication services machining manufacturing services milling services prototyping services turning services',

]

**Results after applying the TFIDF vectorizer:**

[[0. 0.14772204 0.18736667 0.14772204 0. 0.

0.14772204 0.18736667 0. 0. 0. 0.11959372

0. 0.18736667 0. 0.11959372 0.18736667 0.

0.0977757 0.14772204 0. 0. 0. 0.

0.29544407 0. 0. 0. 0.18736667 0.58665421

0.18736667 0.18736667 0.18736667 0.18736667 0. 0.18736667

0. 0. 0.18736667 0.14772204 0.18736667 0.

0. ]

[0.11202235 0.08831971 0. 0.08831971 0. 0.11202235

0.08831971 0. 0.35327882 0.08831971 0.11202235 0.07150241

0.11202235 0. 0.11202235 0.14300483 0. 0.11202235

0.05845791 0. 0.08831971 0.17663941 0.08831971 0.07150241

0.17663941 0.11202235 0.07150241 0.08831971 0. 0.7599528

0. 0. 0. 0. 0.08831971 0.

0.11202235 0.08831971 0. 0.08831971 0. 0.07150241

0.11202235]

[0. 0. 0. 0. 0.27392614 0.

0. 0. 0. 0. 0. 0.

0. 0. 0. 0.17484351 0. 0.

0.14294602 0.21596651 0. 0. 0. 0.17484351

0. 0. 0.17484351 0. 0. 0.85767614

0. 0. 0. 0. 0. 0.

0. 0. 0. 0. 0. 0.17484351

0. ]

[0. 0. 0. 0. 0. 0.

0. 0. 0.3778817 0.18894085 0. 0.1529639

0. 0. 0. 0. 0. 0.

0.12505801 0. 0.18894085 0.3778817 0.18894085 0.1529639

0. 0. 0.1529639 0.18894085 0. 0.62529006

0. 0. 0. 0. 0.18894085 0.

0. 0.18894085 0. 0. 0. 0.1529639

0. ]]

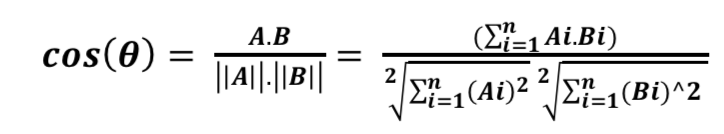
**6. Calculating the Similarity Score using Cosine Similarity:**

In this step we calculate the similarity score between the Input and the capabilities of each manufacturer using the cosine similarity matrix.

**Cosine Similarity:**

Cosine similarity is the measure of similarity between two vectors, by computing the cosine of the angle between two vectors projected into multidimensional space. It can be applied to items available on a dataset to compute similarity to one another via keywords or other metrics. Similarity between two vectors (A and B) is calculated by taking the dot product of the two vectors and dividing it by the magnitude value as shown in the equation below. We can simply say that the CS score of two vectors increases as the angle between them decreases.

Formula for calculating the cosine similarity



**Top 10 Companies with high similarity score:**

company\_score ID

0.445438 381

0.429004 318

0.425167 445

0.390867 463

0.386685 546

0.375450 428

0.365071 690

0.362529 611

0.353399 47

0.353399 1302

**7.Results:**

Based on the Similarity score we will display the manufacturers as suggested manufactures to the user for the entered capabilities, in or case the below companies are the suggested companies for input ["COVID-19 Response"]

