INSY 5377 WEB AND SOCIAL ANALYTICS

FINAL PROJECT REPORT

Explore and Visualize LinkedIn Network using Python and Sentiment Analysis

SHIV NARAIN	1001709882
OSAMA IMTIAZ	1001722863
PAVAN KUMAR NADIPELLI	1001773422

INTRODUCTION

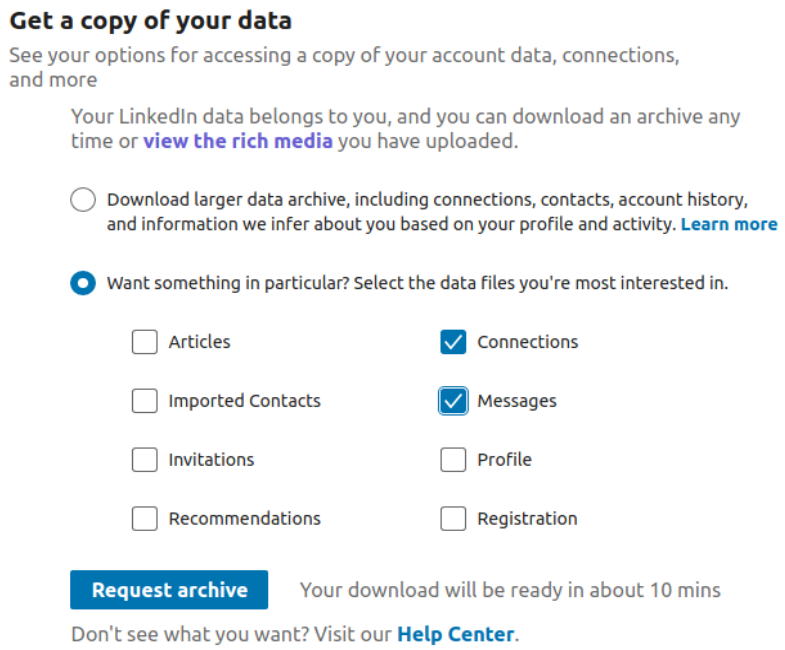
Looking to optimize LinkedIn Profile? Why not make the data work for you? As an active user on LinkedIn with more than 500 connections, we were curious about the statistics of people in Shiv's network as well as the messages he received over the last 2 years.

What is good about analysing LinkedIn data? Well, if you are optimizing your LinkedIn profile for job opportunities, why not use available data as your tool? In this project, we will combine data visualization and natural language processing to analyse Shiv Narain's network and messages. After this project, you should be able to analyse your own LinkedIn data and gain insights from it.

DATA DESCRIPTION AND SOURCE

- We had utilised Shiv Narain's LinkedIn profile data for our project

Here's how you can access it:



We specifically imported the messages and connections data.

Python framework has been used publishing and sharing viz.

OBJECTIVE

Q1) If you are working to expand your connections in the data science world, do most of the people in your network work in a data science-related field?

Q2) How about your messages? Are they mostly positive and about the topics that are related to your interests?

Q3) Which organizations do the people in my network work at?

Q4) What is the sentiment of my LinkedIn messages? I would guess most of them are mostly positive but by what percentage

Q5) What are the negative messages and what words made them negative?

Q6) What are the indicators of positive words?

Q7) How can Shiv optimize his LinkedIn profile and make data work.

METHODOLOGY

Message Analysis

- We first download our personal data from LinkedIn

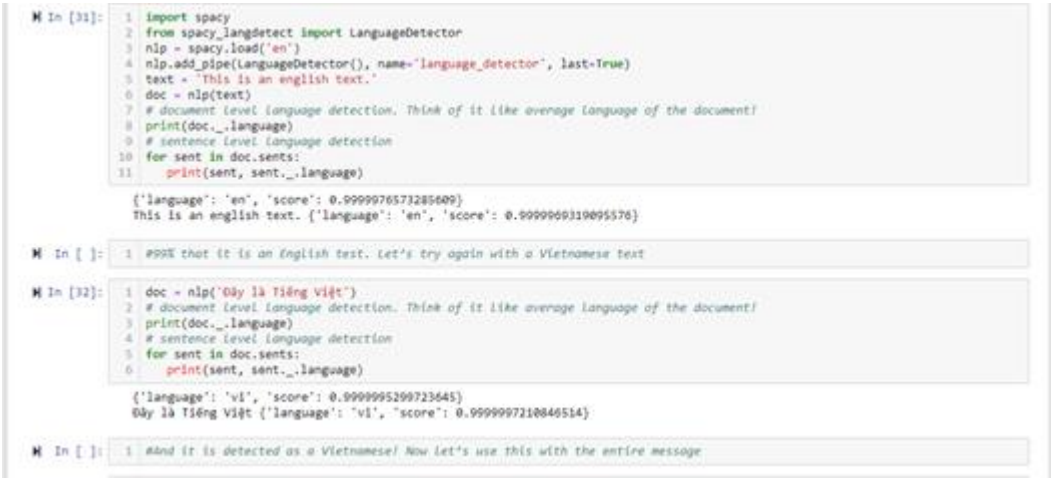
```
In [49]:
import pandas as pd
import numpy as np

messages = pd.read_csv('messages.csv')
messages.head(10)
```

Out[49]:

	CONVERSATION ID	CONVERSATION TITLE	FROM	SENDER PROFILE URL	TO	DATE	SUBJECT	CONT
0	YjQ5MjRjZDUtMjAzMi00YWQ0LTk5MjYtZWQ2MDZkZjg2...	NaN	Shiv Narain	https://www.linkedin.com/in/shivnarain28	Ronnie Dasgupta	2020-12-02 20:18:11 UTC	NaN	Hovi pri
1	YjQ5MjRjZDUtMjAzMi00YWQ0LTk5MjYtZWQ2MDZkZjg2...	NaN	Shiv Narain	https://www.linkedin.com/in/shivnarain28	Ronnie Dasgupta	2020-12-02 20:17:47 UTC	NaN	Yes Kin me ki you a
2	YjQ5MjRjZDUtMjAzMi00YWQ0LTk5MjYtZWQ2MDZkZjg2...	NaN	Ronnie Dasgupta	https://www.linkedin.com/in/ronniedasgupta	Shiv Narain	2020-12-02 20:16:22 UTC	NaN	Oka ver , com so I
3	YjQ5MjRjZDUtMjAzMi00YWQ0LTk5MjYtZWQ2MDZkZjg2...	NaN	Shiv Narain	https://www.linkedin.com/in/shivnarain28	Ronnie Dasgupta	2020-12-02 20:15:34 UTC	NaN	N tho vi ass

- Then detecting the language from the content, and for our requirement we have only filtered out English sentences to use for further analysis.



```
In [31]:
import spacy
from spacy_langdetect import LanguageDetector
nlp = spacy.load('en')
nlp.add_pipe(LanguageDetector(), name='language_detector', last=True)
text = 'This is an english text.'
doc = nlp(text)
# document level language detection. Think of it like average language of the document!
print(doc._.language)
# sentence level language detection
for sent in doc.sents:
    print(sent, sent._.language)

{'language': 'en', 'score': 0.9999976573285609}
This is an english text. {'language': 'en', 'score': 0.9999969319095576}

In [ ]:
#99% that it is an English text. Let's try again with a Vietnamese text

In [32]:
doc = nlp('Đây là Tiếng Việt')
# document level language detection. Think of it like average language of the document!
print(doc._.language)
# sentence level language detection
for sent in doc.sents:
    print(sent, sent._.language)

{'language': 'vi', 'score': 0.9999995299723645}
Đây là Tiếng Việt {'language': 'vi', 'score': 0.9999997210846514}

In [ ]:
#And it is detected as a Vietnamese! Now let's use this with the entire message
```

- Then, we perform entity recognition that clearly makes a distinction between different types of entities. This aids as the most important step to make advanced level analysis. We can see that entities like people names, date or products can be easily differentiated

```
In [34]:
import spacy
from spacy import displacy
from collections import Counter
import en_core_web_sm
from pprint import pprint

#Load model
nlp = spacy.load("en_core_web_sm")

message1 = nlp(messages[154
                        ])

pprint([(word, word.label_) for word in message1.ents])

[(Sahar, 'PERSON'), (F1, 'PRODUCT'), (3 years, 'DATE'), (Shiv Narain, 'PERSON')]

In [63]:
displacy.serve(message1, style='ent')

C:\ProgramData\Anaconda3\lib\runpy.py:193: UserWarning: [W011] It looks like you're calling displacy.serve from within a Jupyt
er notebook or a similar environment. This likely means you're already running a local web server, so there's no need to make
displaCy start another one. Instead, you should be able to replace displacy.serve with displacy.render to show the visualizati
on.
  "__main__", mod_spec)

Hi Sahar PERSON . Thanks for the update. I would not require visa sponsorship for working as a Full time employee for this role as my current F1
PRODUCT visa status permits me to work for 3 years DATE . Hoping to hear from you. Thanks, Shiv Narain PERSON
```

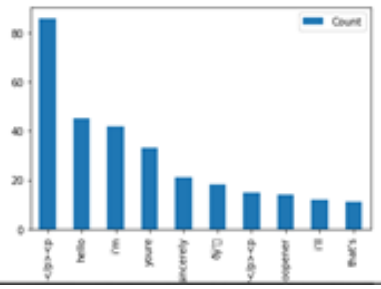
- We have made a list of all the commonly used words. This will give a sense of the way we communicate and the responses we may receive. We can also use a word cloud here to visualize those words but for we thought this was not imperative in this step

```
How many most common words to print: 10

OK. The 10 most common words are as follows

hello :  45
i'm :  42
youre :  33
sincerely :  21
ðy"Ø :  18
shiv</p><p :  15
rel=nooopener :  14
i'll :  12
that's :  11
```

Out[36]: <matplotlib.axes._subplots.AxesSubplot at 0x16a7e3289b0>



- Sentiment Analysis will then be performed to measure the extent of different sentiments that show up in our regular messages.

In [38]:

```
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
nltk.download('vader_lexicon')

sentence = 'I love this weather'
sid = SentimentIntensityAnalyzer()
sid.polarity_scores(sentence)
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     C:\Users\19724\AppData\Roaming\nltk_data...
[nltk_data]   Package vader_lexicon is already up-to-date!
```

Out[38]: {'neg': 0.0, 'neu': 0.323, 'pos': 0.677, 'compound': 0.6369}

SentimentIntensityAnalyzer analyses what percentage are neutral, negative, and positive. Compound is the final result of a combination of percentages. Thus, we could create the function to analyze sentiment based on compound

Results of Sentiment Analysis:

- The below Jupyter cell shows the compounded sentiments of some conversations.

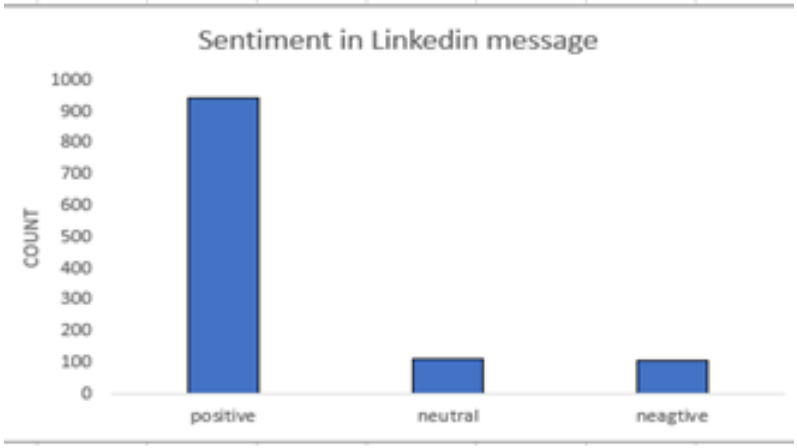
```
In [71]:
s1=messages['CONTENT']
s2=s1[0:11]
for zed in range(10):
    print(s2[zed],   predict_sentiment(s2[zed]))
```

```
How is the hiring process like?  ('Positive', 33.3)
Yes sure. Kindly let me know if you need any further details ('Positive', 47.7)
Okay, I'm very new at the company so I'm not sure what channels I'm supposed to go through but let me see if what I can do ('N
eutral', 88.7)
No, not yet. I thought I will you ask your first and then move ahead.  ('Negative', 15.5)
Have you sent your info in yet? ('Neutral', 100.0)
Sure ('Positive', 100.0)
Okay let me check it out ('Positive', 27.5)
Client Analyst is the job posting under the careers section.  You can find it in this link : https://twocircles.com/us-en/the-
academy/ ('Neutral', 100.0)
Hey Shiv can you send me the posting? ('Neutral', 100.0)
Hi Ronnie.  Hope you are doing good. I came across a  suitable job posting at Two Circles and would like to apply for the sam
e. Would you be open to referring me for the same?  Looking forward to chatting with you.  Thanks, Shiv ('Positive', 23.2)
```

finding sentiment for entire message content

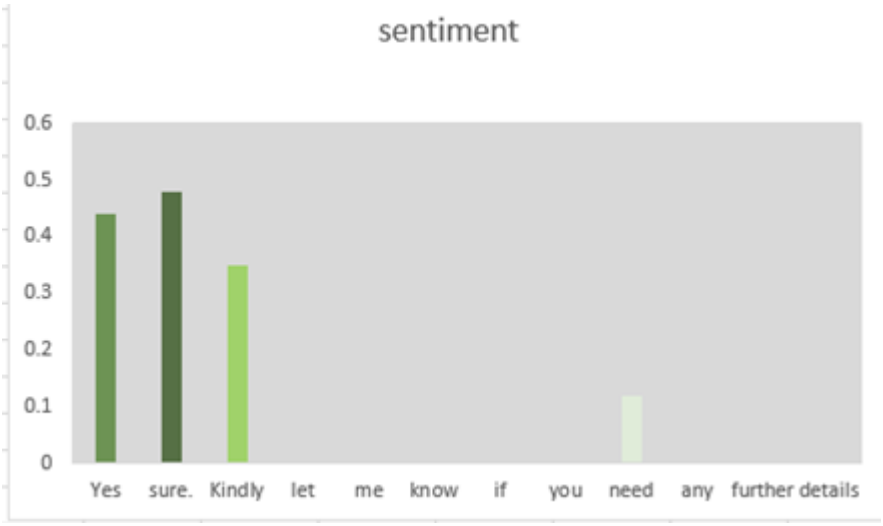
```
In [91]:
import plotly.express as px
for ii in range(0,1161):
    hh= predict_sentiment(s1[ii])
```

- After the compounded sentiments are calculated, we can categorize each sentence into one of the following three categories.



- As expected, most of the messages are positive. But I cannot recall an instance where my messages were negative.
- So, what were these negative messages and what words made them negative?

- Looks like, the negative sentiment is attributed to most of the sentences because of the use of some common words like NO and NOT. There could be other factors as well.
- Also, the words like Yes, Sure, Kindly, etc gives positive sentiments to the sentences.



- This way we can conduct sentiment analysis to the entire messages and use the insights further to segment entities or people based on the sentiments.

NETWORK ANALYSIS

- Start with import and check the data

9	Luis Carlos	Villaseñor	NaN	New Orleans Pelicans	Business Intelligence Intern	15-Nov-20

- Following plot shows that there is a peak in number of new connections per day, especially from January 2020 to July 2020, the period when Shiv was the most active on LinkedIn.

- **Companies** – Which organizations do people in Shiv's network work at? We use a Bar Plot to visualize the same.

- But maybe tree map does a better job to visualize companies in this case? Tree maps display hierarchical data as a set of **nested rectangles**. Each group is represented by a rectangle, which area is proportional to its value.

Python Code:

```
4 fig.show()
```

With a tree map, it is easier to compare the proportion of one company related to the others. It looks like a large percentage of the people from Shiv's network are from University of Texas at Arlington (48). The second-largest percentage is from Amazon (12).

- **Positions** – Here we analyse the positions of different people in the network.

Wow. I did not expect to see so many Data Analysts in the network. It is great that the top common positions in my network are my target groups for networking. Some people might have titles start with 'data scientist' but also have more words in their titles.

Looking at all the positions with words starting with 'Data':

```
>> network.Position.str.startswith('Data').sum()
```

42

"There are 42 data professionals in the network."

- In order to represent the Positions of my network, we use WordCloud representation.

```
24
25      return fig
```

Positions involving words like Data, Engineer, Analyst are higher in number compared to other ones as shown.

Conclusion

- From the above analysis, we can see that Shiv is going in the right direction when it comes to making professional connections. The higher proportion of the network is of data professionals.
- Most people Shiv has connections with work at tech firms excluding UT Arlington which is obvious to show up here.

Additional thoughts:

However, this is just message and network analysis and we have gathered good insights. During the project we have mostly focused on answering our research questions that we assumed were pertinent. But we can also go further and perform some advanced analytics methods such as prescriptive analysis to find out making some changes in the way we network; we can bring in the desired results.