Data Analytics and Visualization End of Semester Project Report

Name:	**Tibabwetiza Muhanguzi**
Reg No:	**2021/HD05/2312U**
Std No:	**2100702312**
Email:	**tibajoel90@gmail.com**

Abstract

This report entails the steps taken within the data analytics exam coursework that was presented on 2nd September. The assignment required the building of a corpus from the given set of comments and classifying a supervisor's comments in different categories. In addition a named entity recognition model was designed in order to take in some of the comments as input and return comments with particular entities. Lastly, corresponding visualizations were prepared using D3js and hosted on an online server for access.

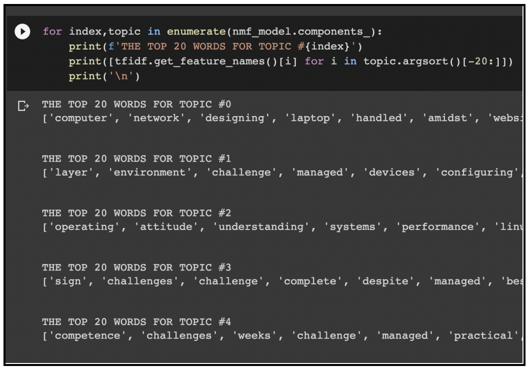
1. Introduction

In order to acquire the comments they were accessed through google colab in a notebook. Having acquired the raw comments, they were then preprocessed using term frequency. Term frequency is acquired through the means of the TfidVectorizer under the sklearn toolkit for machine learning. It facilitated acquiring how often a given word frequently appears within the created corpus across the different comments. It is at this stage that stop words were also eliminated from the comments set using inbuilt stop words. The first approach was to use Non Negative Matrix Factorization after which Vader Analysis was used as will be explored within the report.

2. Non Negative Matrix Factorization (NMF)

This is an unsupervised machine learning algorithm that performs dimensionality reduction as well as text classification. It is often used when a large number of attributes is provided and also works effectively on attributes with weak predictability[1].

Having established that the total number of clusters to be established was 5, an NMF model was created with 5 clusters and the top 20 words within a given cluster were established as well as shown in the figure 2.1 below. Having established that, the clusters were assigned metric scores of 1 to 5. These represented the different target clusters of very poor, poor, neutral, good and excellent[1].

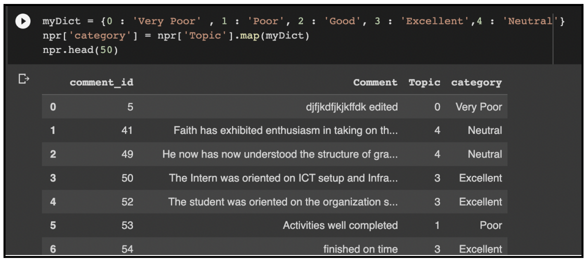


```
for index,topic in enumerate(nmf_model.components_):
    print(f'THE TOP 20 WORDS FOR TOPIC #{index}')
    print([tfidf.get_feature_names()[i] for i in topic.argsort()[-20:]])
    print('\n')
```

```
THE TOP 20 WORDS FOR TOPIC #0
['computer', 'network', 'designing', 'laptop', 'handled', 'amidst', 'webs


THE TOP 20 WORDS FOR TOPIC #1
['layer', 'environment', 'challenge', 'managed', 'devices', 'configuring',


THE TOP 20 WORDS FOR TOPIC #2
['operating', 'attitude', 'understanding', 'systems', 'performance', 'linu


THE TOP 20 WORDS FOR TOPIC #3
['sign', 'challenges', 'challenge', 'complete', 'despite', 'managed', 'bes


THE TOP 20 WORDS FOR TOPIC #4
['competence', 'challenges', 'weeks', 'challenge', 'managed', 'practical',
```

Figure 2.1: Topic Top 20 words

In order to map the different clusters to the scores, this step had to be done by observation as there was no clear score metric to assign to the different clusters for the different comments in the created corpus. This made the method very ambiguous which was a clear limitation in using the NMF method. Below is a figure 2.2 of the classes assigned to the different comments[2].



```
myDict = {0 : 'Very Poor' , 1 : 'Poor', 2 : 'Good', 3 : 'Excellent',4 : 'Neutral'}
npr['category'] = npr['Topic'].map(myDict)
npr.head(50)
```

	comment_id	Comment	Topic	category
0	5	djfjkdfjkjkffdk edited	0	Very Poor
1	41	Faith has exhibited enthusiasm in taking on th...	4	Neutral
2	49	He now has now understood the structure of gra...	4	Neutral
3	50	The Intern was oriented on ICT setup and Infra...	3	Excellent
4	52	The student was oriented on the organization s...	3	Excellent
5	53	Activities well completed	1	Poor
6	54	finished on time	3	Excellent

Figure 2.2: Topic and Category Assignment

The pie chart in figure 2.3 shows the different clusters assigned for the different target classes for the comments. But due to the limitations explained regarding the NMF method[2], it was decided to use the Vader Analysis method from the Natural Language ToolKit.

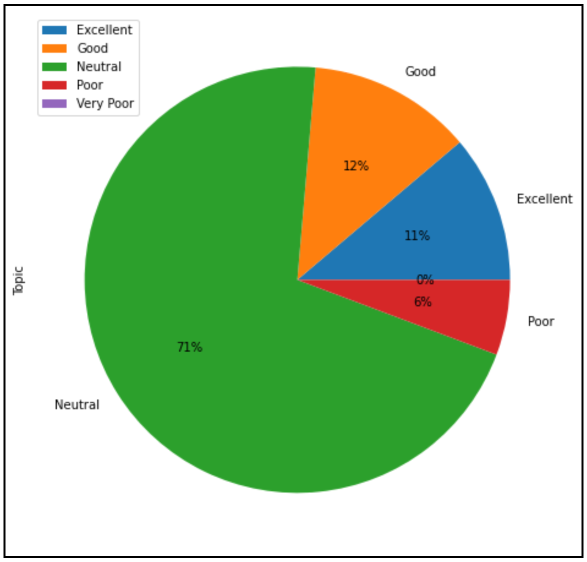


Figure 2.3: Comment Distribution on a pie chart

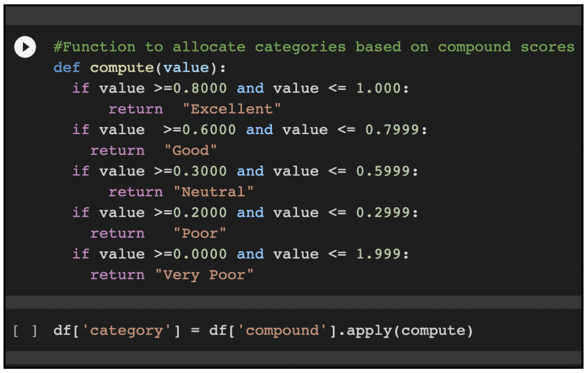
3. VADER Analysis from NLTK

VADER which stands for Valence Aware Dictionary for Sentiment Reasoning is a very powerful tool for establishing text sentiment analysis[3].

The main advantages with it is that it is sensitive to polarity for both negative and positive comments and most desirable is the fact that it captures the strength of emotions in metric format[3].

To use this method, the comments dataset was loaded and preprocessed by removing stop words, repeated sentences as well as empty spaces within some comment columns.

Afterwards, Vader was applied to the corpus and scores were established for the different emotions as well compound scores. A function to categorize the scores according to different cluster names was created as shown in the figure 3.1 below.

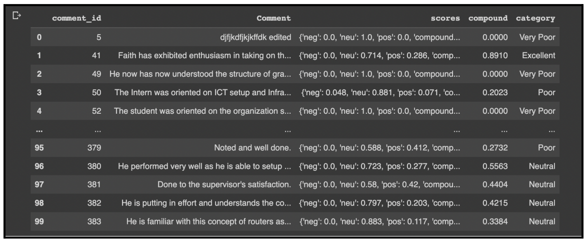


```
#Function to allocate categories based on compound scores
def compute(value):
  if value >=0.8000 and value <= 1.000:
      return  "Excellent"
  if value  >=0.6000 and value <= 0.7999:
    return  "Good"
  if value >=0.3000 and value <= 0.5999:
      return "Neutral"
  if value >=0.2000 and value <= 0.2999:
    return   "Poor"
  if value >=0.0000 and value <= 1.999:
    return "Very Poor"
```

```
[ ]  df['category'] = df['compound'].apply(compute)
```

Figure 3.1: Score based Category Assignment

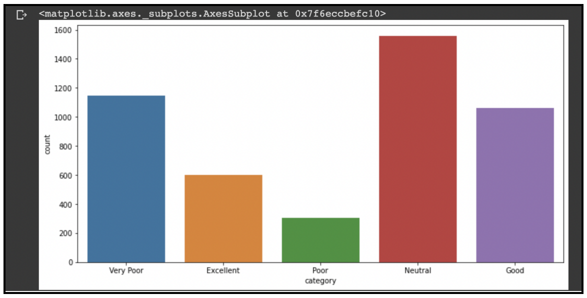
The figure 3.2 below shows the scores with the corresponding categories.



	comment_id	Comment	scores	compound	category
0	5	djfjkdfjkjkffdk edited	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	Very Poor
1	41	Faith has exhibited enthusiasm in taking on th...	{'neg': 0.0, 'neu': 0.714, 'pos': 0.286, 'comp...	0.8910	Excellent
2	49	He now has now understood the structure of gra...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	Very Poor
3	50	The Intern was oriented on ICT setup and Infra...	{'neg': 0.048, 'neu': 0.881, 'pos': 0.071, 'co...	0.2023	Poor
4	52	The student was oriented on the organization s...	{'neg': 0.0, 'neu': 1.0, 'pos': 0.0, 'compound...	0.0000	Very Poor
...	...	...	...	...	...
95	379	Noted and well done.	{'neg': 0.0, 'neu': 0.588, 'pos': 0.412, 'comp...	0.2732	Poor
96	380	He performed very well as he is able to setup ...	{'neg': 0.0, 'neu': 0.723, 'pos': 0.277, 'comp...	0.5563	Neutral
97	381	Done to the supervisor's satisfaction.	{'neg': 0.0, 'neu': 0.58, 'pos': 0.42, 'compou...	0.4404	Neutral
98	382	He is putting in effort and understands the co...	{'neg': 0.0, 'neu': 0.797, 'pos': 0.203, 'comp...	0.4215	Neutral
99	383	He is familiar with this concept of routers as...	{'neg': 0.0, 'neu': 0.883, 'pos': 0.117, 'comp...	0.3384	Neutral

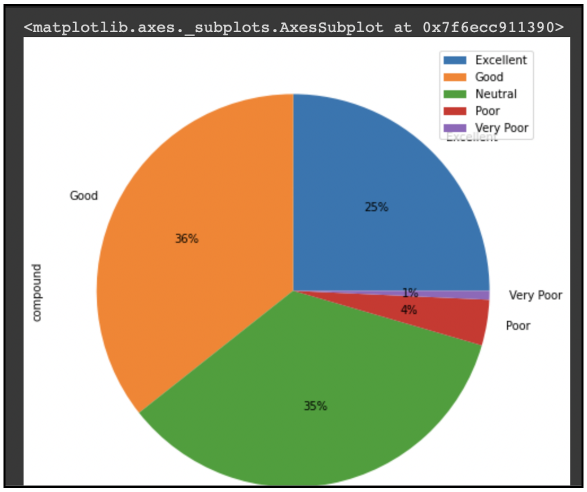
Figure 3.2: Category Assignment based on compound score

The bar graph below in figure 3.3 and the pie chart represented in figure 3.4 the distribution of the different cluster names.



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6eccbefc10>
```

Figure 3.3: Bar Chart with Vader Scores



```
<matplotlib.axes._subplots.AxesSubplot at 0x7f6ecc911390>
```

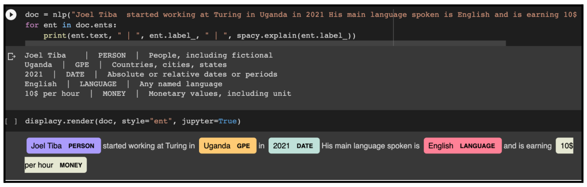
Figure 3.4: Pie chart with Vader Scores

From this it was established that the Vader Method was performing much better than the NMF method[3].

4. Named Entity Recognition

In order to perform the Named Entity Recognition, the spacy model was used for this purpose. It is a pretrained model with specific name entities that are predefined and can be applied to given statements in order to return the corresponding entity names.

It was established that in order to acquire better performance from this model, the sentence subjected to it had to have context in order to enable it to categorize the entities accordingly as shown in the figure 4.1 below. It shows the entities for person, countries, date, language as well as money.



```
doc = nlp("Joel Tiba  started working at Turing in Uganda in 2021 His main language spoken is English and is earning 10$
for ent in doc.ents:
    print(ent.text, " | ", ent.label_, " | ", spacy.explain(ent.label_))
```

```
Joel Tiba    |  PERSON  |  People, including fictional
Uganda  |  GPE  |  Countries, cities, states
2021  |  DATE  |  Absolute or relative dates or periods
English  |  LANGUAGE  |  Any named language
10$ per hour  |  MONEY  |  Monetary values, including unit
```

```
displacy.render(doc, style="ent", jupyter=True)
```

Joel Tiba PERSON started working at Turing in Uganda GPE in 2021 DATE His main language spoken is English LANGUAGE and is earning 10$ per hour MONEY

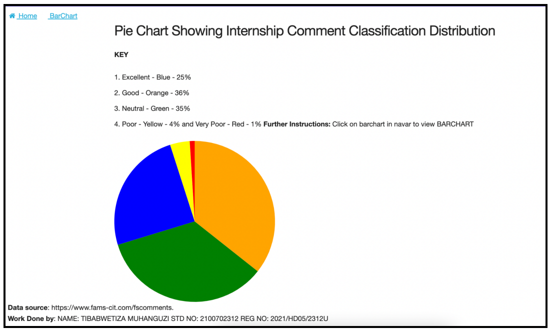
Figure 4.1: Spacy NER in action

This same method was applied for a couple of comments with particular specified entity names which were TIME. PERSON, LOC, ORG and if any of the comments satisfied the entity requirements, then they were returned in a manner similar to the one shown in the figure 4.1 above.

5. Visualizations in D3JS

The visualizations were performed and hosted on a server. Below are images of the visualizations developed for the Vader Analysis method and hosted.

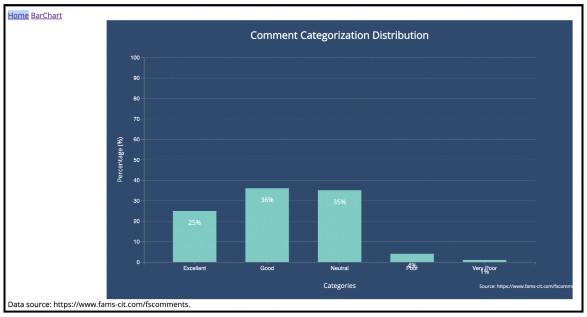
The first visualization is a pie chart showing the comment classification distribution for the corpus.



Data source: https://www.fams-cit.com/fscomments.
Work Done by: NAME: TIBABWETIZA MUHANGUZI STD NO: 2100702312 REG NO: 2021/HD05/2312U

Figure 5.1: d3 js pie chart[4]

A corresponding bar chart was also established to represent the distribution of the comments assigned.



Data source: https://www.fams-cit.com/fscomments.

Figure 5.2: d3 js bar chart[5]

References

[1]S. Kumar, “Unsupervised text classification in python,” *Home*, Aug. 08, 2020. https://www.herevego.com/unsupervised-text-class-python/ (accessed Sep. 16, 2022).

[2]rajiv aiml, “Automatic Ticket Classification Using NMF,” *Kaggle*, Oct. 03, 2021. Accessed: Sep. 16, 2022. [Online]. Available: https://www.kaggle.com/code/rajivaiml/automatic-ticket-classification-using-nmf

[3]A. Beri, “SENTIMENTAL ANALYSIS USING VADER,” *Towards Data Science*, May 27, 2020. Accessed: Sep. 16, 2022. [Online]. Available:

https://towardsdatascience.com/sentimental-analysis-using-vader-a3415fef7664

[4]“Text Analysis Exam.” https://runautomations.com/analyticsexam/index.html (accessed Sep. 16, 2022).

[5]“Bar chart with D3.js.” https://runautomations.com/analyticsexam/barchart.html (accessed Sep. 16, 2022).