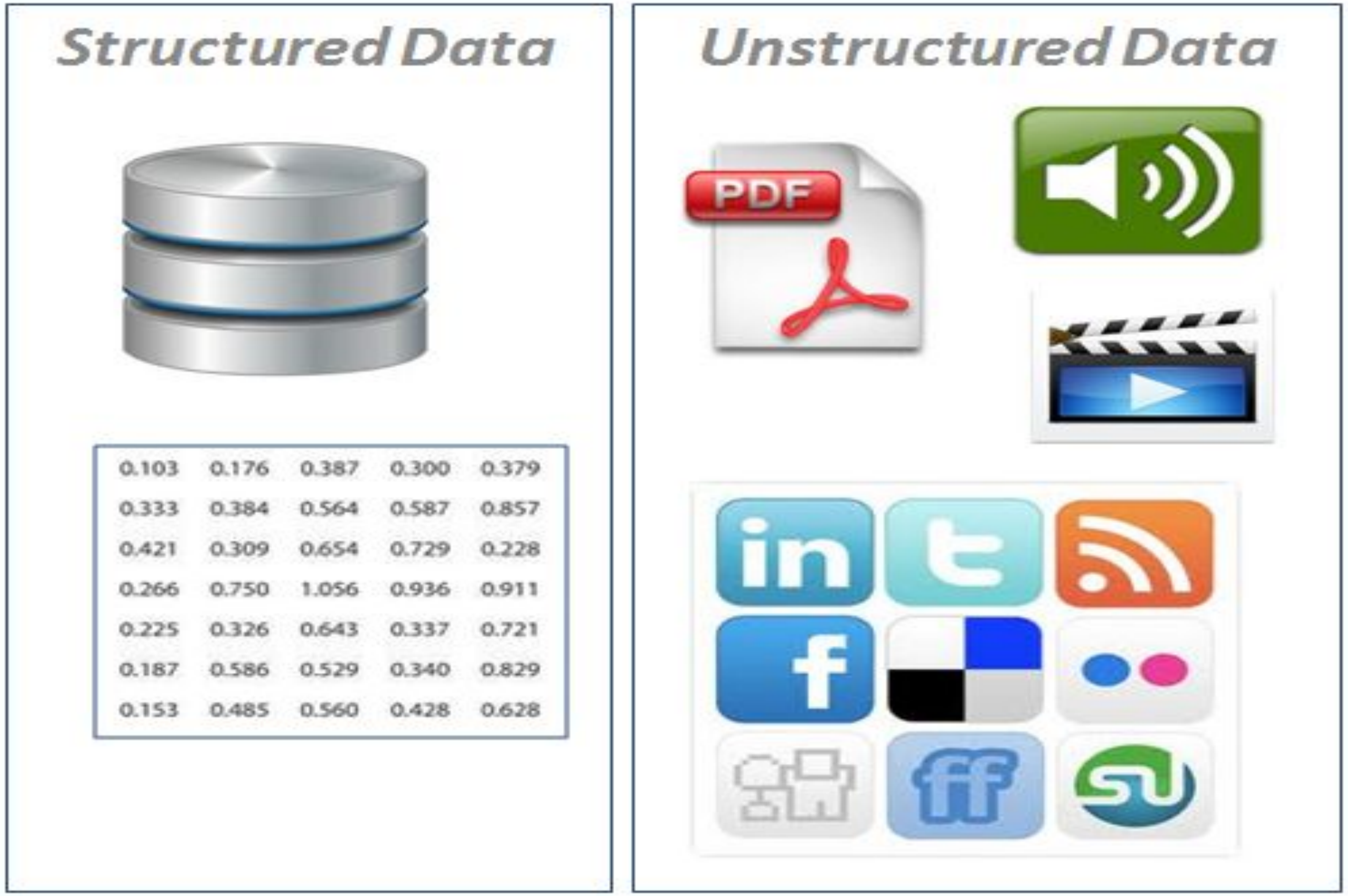
Text Mining and NLP

Contents

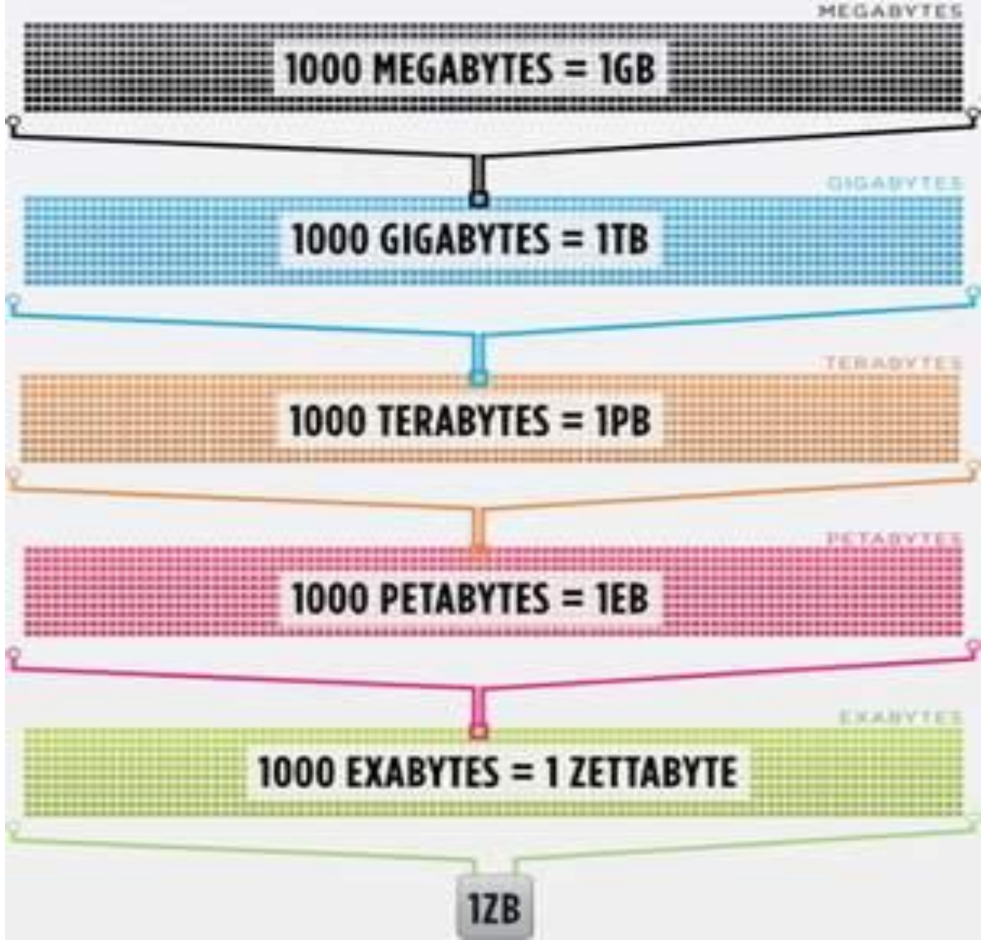
1. Structured and Unstructured Data
2. Features of Unstructured Data
3. What is Content Analysis ?
4. What is Text Analysis ?
5. Case Study
6. Text Mining in Python
7. Word Cloud in Python
8. Text Mining Using Matplotlib

Structured Vs. Unstructured Data



0.103	0.176	0.387	0.300	0.379
0.333	0.384	0.564	0.587	0.857
0.421	0.309	0.654	0.729	0.228
0.266	0.750	1.056	0.936	0.911
0.225	0.326	0.643	0.337	0.721
0.187	0.586	0.529	0.340	0.829
0.153	0.485	0.560	0.428	0.628

Unstructured Data Growth



- Research from IDC (International Data Corporation) shows that unstructured content accounts for 95% of all digital information, with estimate of compound annual growth at 65%

- By 2020, IDC predicts the volume of digital data will have reached 40,000 EB or 40 ZB

Features Of Unstructured Data

Does not reside in traditional databases and data warehouses

May have an internal structure, but does not fit a relational data model

Generated by both humans and machines

- Textual and social media content
- Machine-to-machine communication

Examples Of Unstructured Data

Examples of unstructured data include:

- **Personal messaging** – Email, instant messages, tweets, chat
- **Business documents** – Business reports, presentations, survey responses
- **Web content** – Web pages, blogs, wikis, audio files, photos, videos
- **Sensor output** – Satellite imagery, geo-location data, scanner transactions

Value Of Unstructured Data

Unstructured data provides a rich source of information about people, households and economies.

- It may enable more accurate and timely measurement of a range of demographic, social, economic and environmental phenomena
  - When combined with traditional data sources
  - As a replacement for traditional data sources
- As a result, it presents unprecedented opportunities for official statistics to
  - Improve delivery of current statistical outputs
  - Create new information products not possible with traditional data sources

What is Content Analysis ?

- For unstructured data to be useful, **it must be analysed to extract and expose the information it contains.**

- Content analysis is used to quantify and analyze the presence, meanings and relationships of certain words, themes, or concepts.

- Different types of analysis are possible, such as:

  - **Entity analysis** – People, organisations, objects and events, and the relationships between them.

  - **Topic analysis** – Topics or themes, and their relative importance.

  - **Sentiment analysis** – Subjective view of a person to a particular topic.

What Is Text Analysis ?

- Text Mining is also known as **Text Data Mining (TDM)** and **Knowledge Discovery in Textual Database (KDT)**

- It is a process of identifying novel information from a collection of texts (Also known as a ‘Corpus’)

- **Corpus is a collection of ‘documents’ containing natural language text.** Here, documents, generally, are sentences. Each document is represented as a separate line.

Search Vs. Discover

	Search (Goal-oriented)	Discover (Opportunistic)
Structured Data	Data Retrieval	Data Mining
Unstructured Data (Text)	Information Retrieval	Text Mining

Case Study – HR Appraisal Process Feedback

Background

- The company XYZ carried out Annual Performance Appraisal process which is a routine HR process.
- The employees were asked to give feedback about the overall process and questions used for assessing their performance level.

Objective

- To understand the employee sentiments and incorporate recommendations in the current performance appraisal process.

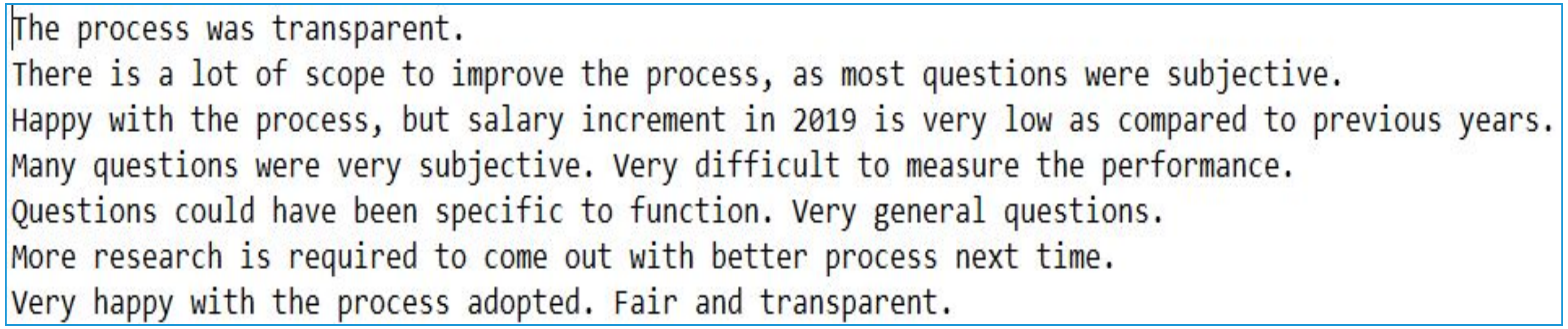
Available Information

- Feedback and comments from the employees were stored in a text document.

Data Snapshot

HR Appraisal process

Text Observations



The process was transparent.
There is a lot of scope to improve the process, as most questions were subjective.
Happy with the process, but salary increment in 2019 is very low as compared to previous years.
Many questions were very subjective. Very difficult to measure the performance.
Questions could have been specific to function. Very general questions.
More research is required to come out with better process next time.
Very happy with the process adopted. Fair and transparent.

*

These are the comments received from employees.
Note that, data is not in structured format.

Text Mining In Python

```
#Install NLTK library in Anaconda Prompt
```

```
pip install nltk
```

```
#Import NLTK library
#Import data and convert into ‘Corpus’
```

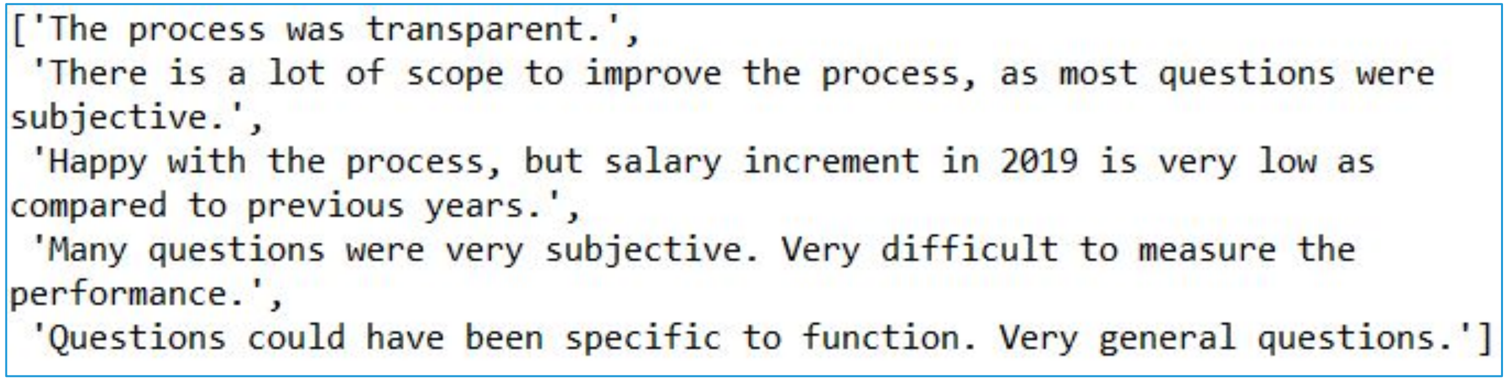
```
import nltk
nltk.download()
from nltk.book import *
text = [line.rstrip() for line in open("HR Appraisal
process.txt")]
text[0:5]
```

❑ Install and load NLTK(Natural Language Toolkit) library.
❑ rstrip() reads all text lines from a file or connection.
❑ rstrip() interprets each element of the vector as a document. It converts and saves data as a corpus.

* Note : When imported nltk, nltk.download() will download the required libraries from NLTK for text mining. Run nltk.download() only for the first time

Text Mining In Python

```
# Output:
```



```
['The process was transparent.',
 'There is a lot of scope to improve the process, as most questions were
subjective.',
 'Happy with the process, but salary increment in 2019 is very low as
compared to previous years.',
 'Many questions were very subjective. Very difficult to measure the
performance.',
 'Questions could have been specific to function. Very general questions.']
```

Interpretation:

□ **text[0:5]** prints first 5 text lines from the data with each line as one set of strings.

```
# Display a particular document from corpus.
```

```
text[2]
```

```
'Happy with the process, but salary increment in 2019 is very low as
compared to previous years.'
```

❑ **text[2]** prints text line of specified number in []. Here it is printing 3rd line.

❑ Python indexing starts from 0, thus 2 represents 3rd data point (sentence).

Text Mining In Python

```
# Clean the Corpus for further analysis
```

```
corp = [item.lower() for item in text]
corp [2]
```

```
'happy with the process, but salary increment in 2019 is very low as
compared to previous years.'
```

```
from string import punctuation
remove_punc = str.maketrans('','', punctuation)
corp = [item.translate(remove_punc) for item in corp]
corp[2]
```

```
'happy with the process but salary increment in 2019 is very low as
compared to previous years'
```

```
from string import digits
remove_digits = str.maketrans('', '', digits)
corp = [item.translate(remove_digits) for item in corp]
corp[2]
```

```
'happy with the process but salary increment in is very low as
compared to previous years'
```

- ❑ **lower()** converts text to lowercase.
- ❑ **maketrans(",", punctuation)** removes punctuation.
- ❑ **maketrans(", ", digits)** removes numbers.

Text Mining In Python

```
# Clean the Corpus for further analysis
```

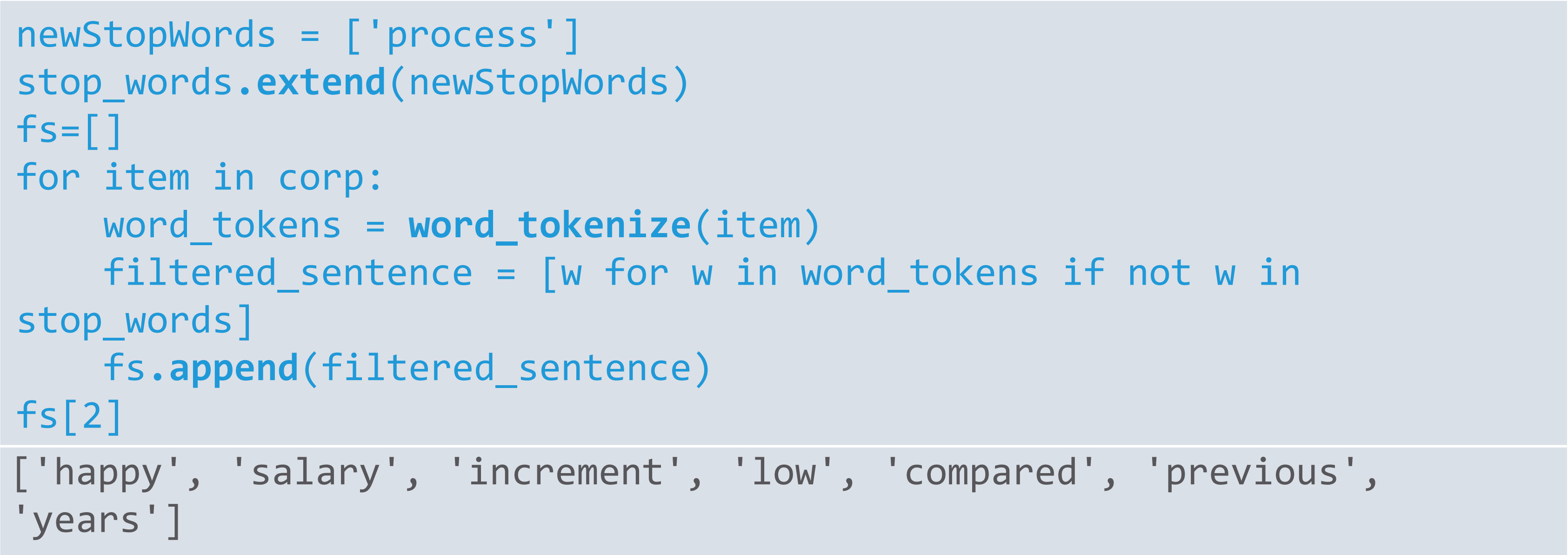
```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
stop_words = nltk.corpus.stopwords.words('english')
fs=[]
for item in corp:
    word_tokens = word_tokenize(item)
    filtered_sentence = [w for w in word_tokens if not w in
stop_words]
    fs.append(filtered_sentence)
fs[2]
```

```
['happy', 'process', 'salary', 'increment', 'low', 'compared’,
'previous', 'years']
```

❑ **stopwords("english")** remove stop words like: i, me, our, and, the, is, etc. There are more than 100 in-built English Stopwords in Python NLTK. Use stopwords("english") to view the list of these stopwords.

Text Mining In Python

```
# Clean the Corpus for further analysis
```



```
newStopWords = ['process']
stop_words.extend(newStopWords)
fs=[]
for item in corp:
    word_tokens = word_tokenize(item)
    filtered_sentence = [w for w in word_tokens if not w in 
stop_words]
    fs.append(filtered_sentence)
fs[2]
```

```
['happy', 'salary', 'increment', 'low', 'compared', 'previous', 
'years']
```

- ❑ If you wish to remove specific words from the corpus use **.extend("word")** to add the word in list of stopwords. Here “**process**” word is removed.

Text Mining In Python

```
# Convert to term-document matrix format
```

```
import itertools
filtered_text = list(itertools.chain.from_iterable(fs))
fdist = nltk.FreqDist(filtered_text)
```

□ **FreqDist()** gives frequency of each word in the list

```
# Find most common words i.e. words having highest frequency
```

```
fdist.most_common(10)
```

```
[('questions', 13),
 ('hr', 12),
 ('happy', 10),
 ('subjective', 8),
 ('fair', 7),
 ('performance', 6),
 ('work', 6),
 ('difficult', 5),
 ('measure', 5),
 ('salary', 4)]
```

Interpretation:

- “questions”, “hr”, “happy”, “subjective”, “fair”, “performance”, “work”, “difficult”, “measure”, ”salary” are the top 10 words by frequency.
- The frequencies of the words are listed besides them.

□ **fdist.most_common(n)** gives the list of top n words sorted highest to lowest by frequency

Word Cloud In Python

Word cloud, as the name suggests, is an **image showing compilation of words**, in which, **size of words indicates its frequency or importance**.

```
# Install the library “wordcloud” in Anaconda Prompt
```

```
pip install wordcloud
```

```
# Get Word Cloud
```

```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
wordcloud =
WordCloud(background_color="white").generate(str(filtered_text))
plt.figure(figsize = (8, 8))
plt.imshow(wordcloud); plt.axis("off")
plt.tight_layout(pad = 0); plt.show()
```

❑ **background.color** allows you to select the color of the background.

❑ **fig.size** allows you to adjust the size/dimensions of the wordcloud.

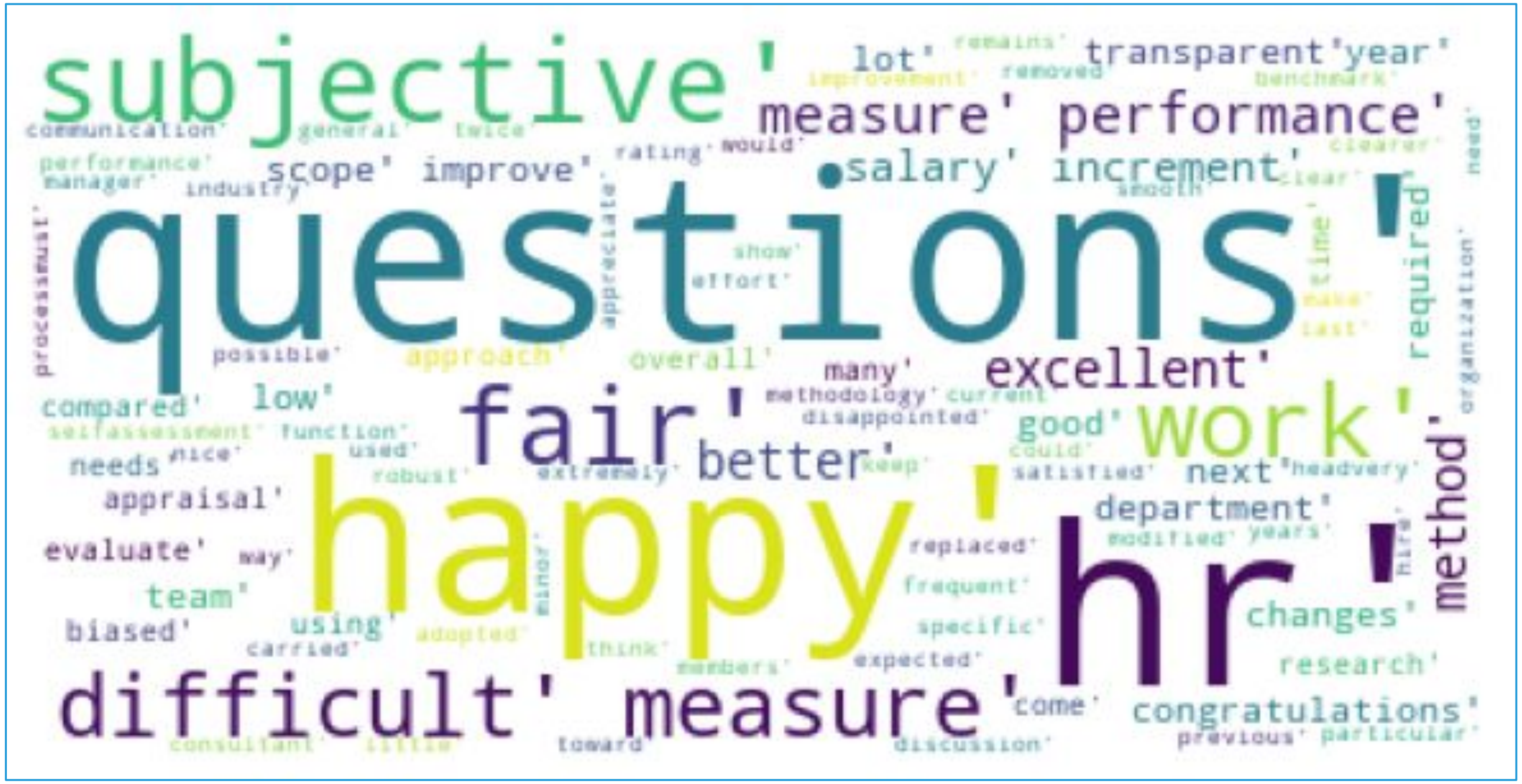
❑ **plt.imshow()** is used to display data as an image.

❑ **plt.axis("off")** means axis lines and labels are turned off.

❑ **plt.tight_layout()** automatically adjusts subplot parameters to give specified padding (**here 0**).

Word Cloud In Python

```
# Output :
```



Interpretation:

- Word ‘questions’ has largest size, indicating most frequent word followed by ‘happy’ and ‘hr’ and so on..

Text Mining Using Matplotlib

```
# Plotting frequent terms as a bar plot
```

```
a = fdist.most_common(10)
```

```
# Transform as a dataframe
```

```
import pandas as pd
b = pd.DataFrame(a)
b = b.rename(columns={0:'Words',1:'Freq'})
```

```
# Horizontal bar plot
```

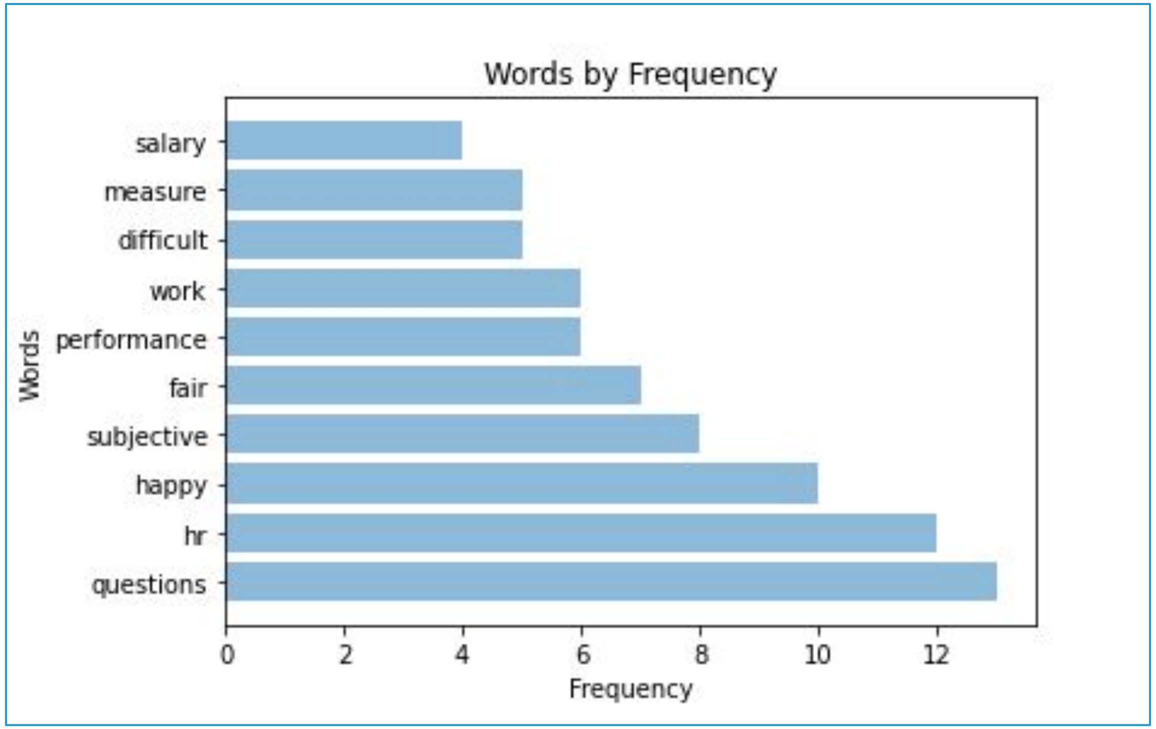
```
import numpy as np
c=b.Words
y=np.arange(len(c))
x=b.Freq

plt.barh(y, x, align='center', alpha=0.5)
plt.yticks(y, c);plt.ylabel('Words')
plt.xlabel('Frequency');plt.title('Words by Frequency')

plt.show()
```

Text Mining Using Matplotlib

```
# Output :
```



Interpretation:

- Graph shows the frequency of the top 10 words by frequency on a horizontal bar graph. “Very” is the most frequent word with frequency 14.

Quick Recap

In this session, we learnt **Text Mining in Python** :

Unstructured Data

- Does not reside in traditional databases and data warehouses.
- Example: emails, tweets, feedback, blogs, webpages, etc.

Text Analysis

- Process of identifying novel information from a collection of texts. (Also known as a 'Corpus')

Text mining in Python

- Install '**nltk**' library. Convert data into corpus.
- Clean the corpus: convert all words to lowercase/uppercase, remove punctuation, numbers, stopwords, words.

Word Cloud in Python

- An image showing compilation of words, in which, size of words indicates its frequency or importance.
- Install '**wordcloud**' library.