21. WordCloud

September 13, 2024

0.0.1 Firsly install wordcloudlibrary

[9]:
```
!pip install wordcloud
```

```
Collecting wordcloud
  Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl.metadata (3.5 kB)
Requirement already satisfied: numpy>=1.6.1 in
c:\users\tanishq\anaconda3\lib\site-packages (from wordcloud) (1.26.4)
Requirement already satisfied: pillow in c:\users\tanishq\anaconda3\lib\site-
packages (from wordcloud) (10.2.0)
Requirement already satisfied: matplotlib in
c:\users\tanishq\anaconda3\lib\site-packages (from wordcloud) (3.8.0)
Requirement already satisfied: contourpy>=1.0.1 in
c:\users\tanishq\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(1.2.0)
Requirement already satisfied: cycler>=0.10 in
c:\users\tanishq\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(0.11.0)
Requirement already satisfied: fonttools>=4.22.0 in
c:\users\tanishq\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(4.25.0)
Requirement already satisfied: kiwisolver>=1.0.1 in
c:\users\tanishq\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(1.4.4)
Requirement already satisfied: packaging>=20.0 in
c:\users\tanishq\anaconda3\lib\site-packages (from matplotlib->wordcloud) (23.1)
Requirement already satisfied: pyparsing>=2.3.1 in
c:\users\tanishq\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in
c:\users\tanishq\anaconda3\lib\site-packages (from matplotlib->wordcloud)
(2.8.2)
Requirement already satisfied: six>=1.5 in c:\users\tanishq\anaconda3\lib\site-
packages (from python-dateutil>=2.7->matplotlib->wordcloud) (1.16.0)
Downloading wordcloud-1.9.3-cp311-cp311-win_amd64.whl (300 kB)
   ---------------------------------------- 0.0/300.2 kB ? eta -:--:--
   -- ------------------------------------- 20.5/300.2 kB 330.3 kB/s eta 0:00:01
   ------- -------------------------------- 61.4/300.2 kB 656.4 kB/s eta 0:00:01
   ---------------- ----------------------- 122.9/300.2 kB 1.0 MB/s eta 0:00:01
```

```
   ------------------ ------------------- 143.4/300.2 kB 950.9 kB/s eta 0:00:01
   -------------------- ----------------- 174.1/300.2 kB 807.1 kB/s eta 0:00:01
   ------------------------ ------------- 204.8/300.2 kB 831.5 kB/s eta 0:00:01
   ------------------------ ------------- 204.8/300.2 kB 831.5 kB/s eta 0:00:01
   -------------------------------- ----- 256.0/300.2 kB 684.6 kB/s eta 0:00:01
   ---------------------------------------- 300.2/300.2 kB 743.4 kB/s eta 0:00:00
Installing collected packages: wordcloud
Successfully installed wordcloud-1.9.3
```

[22]:
```
# Create a list of word
input_words = ("innovation, technology, data, science, future, learning,␣
 ↪research, development, analytics, growth, solutions, creativity, discovery,␣
 ↪progress, insight, collaboration, trends, advancement, success, strategy")
```

[24]:
```
input_words
```

[24]:
```
'innovation, technology, data, science, future, learning, research, development,
analytics, growth, solutions, creativity, discovery, progress, insight,
collaboration, trends, advancement, success, strategy'
```

import the WordCloud class from the wordcloud library.

import the pyplot module from the matplotlib library.

[15]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

[26]:
```
# Create the wordcloud object
wordcloud = WordCloud(width=480, height=400, margin=2).generate(input_words)
```

[28]:
```
# Display the generated image:
plt.imshow(wordcloud, interpolation='bicubic')
plt.axis('off')
plt.margins(x=0, y=0)
plt.show()
```

[33]:
```
input_words2 = ("imagination, art, design, inspiration, colors, creativity,␣
↪expression, passion, style, vision, innovation, aesthetics, ideas,␣
↪originality, beauty, dynamic, conceptual, technique, craft, style")
```

[47]:
```
wordcloud2 = WordCloud(width=700, height=500, margin=50).generate(input_words2)
```

[67]:
```
plt.imshow(wordcloud2, interpolation='bicubic')
# interpolation='bicubic' -- This method uses the 16 nearest pixels (a 4x4␣
↪grid) to estimate the color of each pixel.
plt.axis('on') # Shows axis
plt.margins(x=20, y=10) # horizontal and vertical margin (padding)
plt.show()
```


passion
originality
vision
innovation
imagination
aesthetics
creativity
beauty
colors
dynamic
ideas
conceptual
art
style
expression
craft
technique
design
inspiration
0
100
200
300
400
500
600