WC = WordCloud(width = 100,height = 50, max\_font\_size = 30, min\_font\_size=60, colormap = ‘’,mask = give\_image\_path )

**parameters used in Word-Cloud**:-

width :- width of the canvas int default(400)

height:- height of the canvas int default(200)

max\_font\_size:- int or None default=None, Maximum font size for the largest word. if none, the height of image is used.

min\_font\_size:- int default =4 , smallest font size to use. will stop when there is no more room in this size.

Colormap:- string or Matplotlib colormap, default='viridis', matplotlib colormap to randomly draw color for each word. ignore if 'color\_fun' is specified.

Mask:- give\_image\_path , # nd-array or None default=None

contour\_width:- float defautl=0, if mask is not None and contour\_width > 0, draw the mask contour.

contour\_color:- color value default= black.

Scale:- float default=1, scaling between computation and drawing. For large word-cloud images, using scale instead of larger canvas size is significantly faster, but might lead to coarser fit for the words.

prefer\_horizontal:- float default(0.09) The ratio of times to try horizontal fitting as opposed to vertical. if prefer\_horizontal < 1, the algoritham will try rotating the word, if it dosen't fit.

font\_step:- int default=1, step size for the font. font\_step>1 might speed up computation but give a worse fit.

max\_words:- number default=200, the maximum number of words.

Stopwords:- set of strings or None, the words that will be eliminated. if None, the built-in STOPWORDS list will be used

background\_color:- color value default='black' , color for word cloud image

mode:- string default='RGB', Transparent backgroup will be generated when mode is 'RGBA' and background color is None.

normalize\_plurals:- bool, default=true, whether to remove traling 's'from words. if true and a word appears with and without a trailing 's' the oen with trailing 's' will removed and its counts are added to the version without trailing 's' --unless the word ends with 'ss'.

color\_func:- callable, defaut=None, callable with parameters word, font\_size, position, orientaiton, font\_path, random\_state that returns a PIL color for each word.