**Synopsis of Functions used**

| **Input()** | Get data from the user. |
| --- | --- |
| **Matplotlib** | |
| **(i) Matplotlib.pyplot.bar()** | Draw bar graph. |
| **(ii) Matplotlib.pyplot.pie()** | Draw pie chart |
| **(iii) Matplotlib.pyplot.plot()** | Draw graph |
| **(iv) Matplotlib.pyplot.figure()** | It indicated size of the chart |
| **(iv) Matplotlib.pyplot.title()** | Display tile for the chart |
| **(V) Matplotlib.pyplot.tick()** | Makes the label in direction |
| **(vi) Matplotlib.pyplot.xlabel()** | Labels the x axis |
| **(vii) Matplotlib.pyplot.ylabel()** | Labels the y axis |
| **(viii) Matplotlib.pyplot.tight\_layout()** | Adjust subplot params so that subplots fit in to the figure |
| **(xi) Matplotlib.pyplot.axis()** | Mark the axis and label |
| **(xii) Matplotlib.pyplot.hist()** | Draw histogram |
| **(x) Matplotlib.pyplot.subplot()** | Draw multiple plots in one figure |
| **(x) Matplotlib.pyplot.grid()** | Display the grid in graph |
| **(x) Matplotlib.pyplot.boxplot()** | Draw boxplot graph |
| **nunique()** | Function returns a series with n number of distinct observations over the requested access. |
| **idmax()** | Returns a series with an index of maximum for each column. |
| Date and time function | |
| **(i) today()** | Extract current date |
| **(ii) extract** | It extracts the year, month, day from column |
| **(iii)Series.dt.date** | date attribute return numpy array of python datetime.date object |
| **(iv)to\_datetime** | convert column into date time |

| **fillna()** | replaces the NULL values with a specified value. |
| --- | --- |
| **count()** | Returns the number of times an object appears. |
| **print()** | Prints what the user specifies in the brackets and quotes |
| **groupby()** | It combines the objects based on certain conditions. |
| **reset\_index()** | Allows to reset index number back to default like 0,1, 2 |
| **max()** | returns maximum value from a list of values or group of values. |
| **get()** | fetch value of key from a dictionary |
| **nlargest()** | used to get the largest n elements. |
| **tolist()** | used to convert the data elements of an array into a list. |
| **stack()** | used to stack prescribed levels from column to index. |
| **count()** | count the list or series |
| **merge()** | Join two tables by using a common column. |
| **pd.to\_datetime** | returns datetime object for a given date. |
| **plot()** | Used to plot lines/graphs. |
| **legend()** | Used to plot legends on the graph axis. |
| **plt.show()** | Used to display figures of graphs. |
| **sort\_values()** | Used to sort the rows of a DataFrame |
| **iloc()** | Used to access data based on its integer position rather than label |
| **tolist()** | Used to convert the data contained in an array, Series, or DataFrame into a standard Python list. |
| **sns.violinplot()** | Used to create a violin plot. A violin plot is a visualization that combines the features of a box plot and a kernel density plot |
| **corr()** | Used to calculate the correlation between columns in a DataFrame |
| **sns.heatmap()** | used to create a heatmap visualization |
| **apply()** | used to apply a given function along the axis of a DataFrame |
| **lamda()** | Lambda functions are particularly useful when you need a quick, short function to be used in a specific context |
| **len()** | used to determine the length or the number of elements in an object. |
| **value\_counts()** | used to count the occurrences of unique values |
| **flatten()** | used to convert a multi-dimensional array or nested list into a one-dimensional array or list |
| **warnings.filterwarnings(‘ignore)** | It suppresses all warnings generated by program code. |
| **fstring** | It is a literal string, prefixed with ‘f’ which contains expressions inside braces. |
| **sns.jointplot** | It will create a scatter plot with two marginal histograms by default. |
| **index** | It helps you find the index position of an element or an item in a string of characters or list of items. |
| **plt.imshow** | It creates an image from a 2-dimensional array. |
| **mean()** | Finds average |
| **pivot\_table()** | creates pivot table in Python |
| **agg()** | Aggregates using one or more operations  over the specified axis |

**Libraries used**

| pandas | This library is used for working with datasets. It contains functions for cleaning, manipulating and analyzing the data. |
| --- | --- |
| numpy | This library is used for working with arrays. |
| matplotlib | It is a data visualization and graphical plotting library. |
| matplotlib.pyplot | It is used for creating static, interactive and animated visuals in Python. |
| seaborn | This library is used for data visualizations for machine learning and data science. |
| plotly.express | It is a built-in part of plotly library and helps in creating common figures,using graph objects. |

**Code Optimization We Implemented**

1. Used numpy arrays instead of lists,this will save memory space and execution time will be reduced.
2. Used generator expressions instead of list comprehensions.
3. Used keys for sorts.
4. For loop optimization arange() has been used.
5. loc() function is used to find the particular position.
6. To search we used sort() and filtered using the index function.
7. Generate\_frequency is used to generate wordcloud.
8. Avoid global variables instead of using local variables.
9. Created an inline function like lambda.