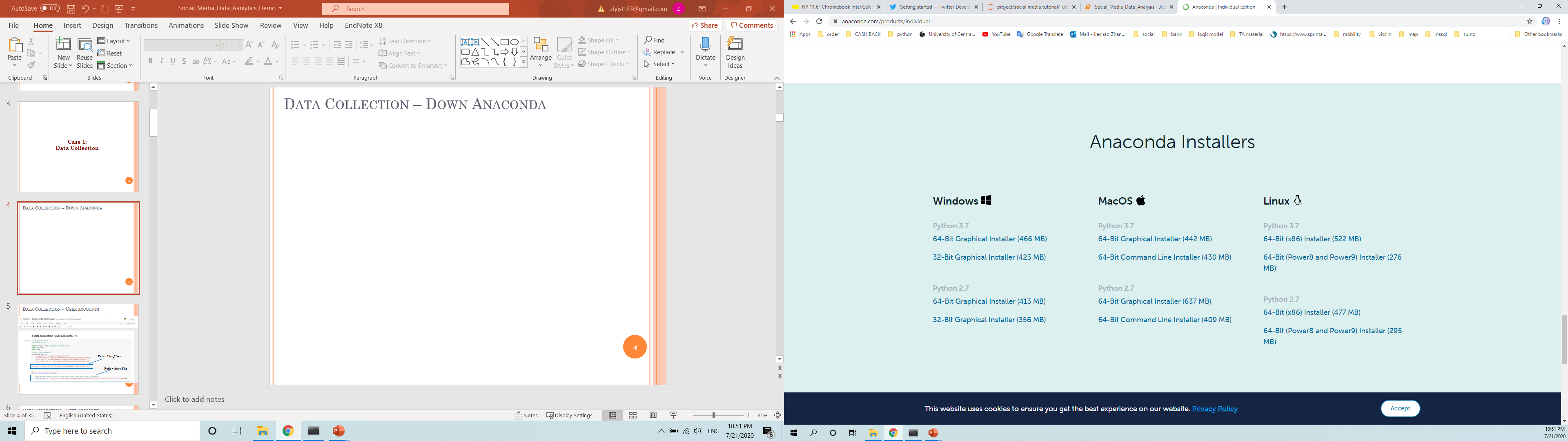
**A Tutorial of Social Media Data Analysis**

This tutorial is an instruction for running the code of social media data analysis. It mainly contains four parts and the codes can be seen as the attached Jupyter Notebook – **‘**Social Media Data Analysis.ipynb**’**. It has four sections: i) data collection, ii) sentiment analysis, iii) visualization of the sentiment analysis results, iv) visualization of the topic analysis results, and iv) topic analysis.

**1. Software Installation and Data Downloading**

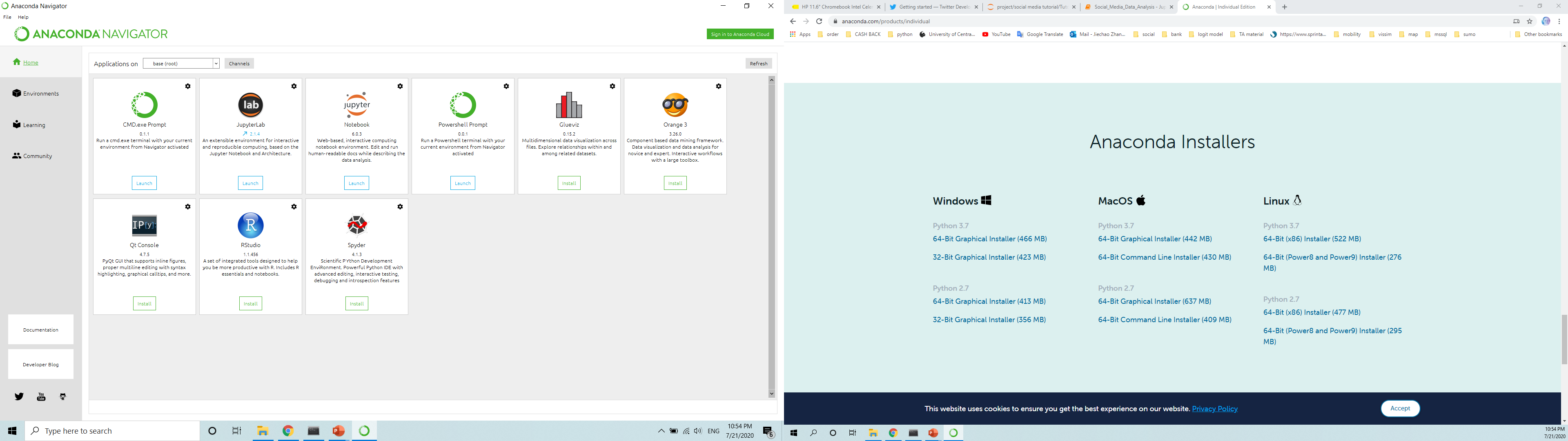
**1.1 Software Installation**

In this project, we use Jupyter notebook as the integrated development environment for Python. To install the Jupyter notebook, we can download the anaconda from the official website, shown in **Figure 1.1**. The link of the website is: https://www.anaconda.com/products/individual. We can choose the suitable version of anaconda from the website (e.g. 64-Bit Graphical Installer).



**Figure 1.1** The official website of Anaconda

After installing the anaconda, we can open the Jupyter notebook from anaconda navigator, shown in **Figure 1.2**.



**Figure 1.2** The platform of anaconda navigator

In the Jupyter notebook, we can update the code file - Social Media Data Analysis.ipynb, shown in **Figure 1.3**.

A screenshot of a computer

Description automatically generated

**Figure 1.3** The platform of Jupyter notebook

**1.2 Data Collection**

1.2.1 Apply for a Twitter Developer Account

To collect the Twitter data, one of the prerequisites is to have a Twitter developer account. Thus, we need to apply for a Twitter developer account online through the following link: https://developer.Twitter.com/en/apply-for-access. **Figure 1.4** shows the website.

A screenshot of a cell phone

Description automatically generated

**Figure 1.4** The website for applying the Twitter developer account

1.2.2 Data Collection (User Accounts)

In the Jupyter Notebook, the code for data collection is shown under the heading ‘**1 Data Collection (User Accounts)**’, **seen as Figure 1.5**.

The **Twitter\_app\_auth** is the Twitter API credentials for which one needs to apply to the Twitter official website.

To collect Twitter data by user accounts, we need a ‘csv’ file called ‘List\_User.csv’ (this file name should be fixed), seen in **Figure 1.6**. The ‘List\_User.csv’ file contains all the user accounts for which data need to be collected, and the format can be seen in **Figure 1.6**. Put the ‘List\_User.csv’ file and the code in the same folder.

For different collection time, we can create different folders to save the Twitter data. Each folder must contain both the ‘List\_User.csv’ file and the data collection code. From the code, we can change the ‘June\_10\_tweets’ (shown in **Figure 1.5**) to the expected date to save the Twitter data with a different file name.

After all the files are prepared, run the code and the Twitter data will be collected in the same folder where the code for data collection is kept.

A screenshot of a cell phone

Description automatically generated

**Figure 1.5** Example of User Account Data Collection Code

A screenshot of a computer

Description automatically generated

**Figure 1.6** The example of List\_User.csv and output file

1.2.3 Data Collection (Keywords)

The code for data collection is shown under the heading ‘**2 Data Collection (keywords)**’, **seen as Figure 1.7 (a)**.

The **Twitter\_app\_auth** is the Twitter API credentials for which one needs to apply to the Twitter official website.

To collect Twitter data by user accounts, we need a ‘csv’ file called ‘0.List\_KW.csv’ (this file name should be fixed), seen as **Figure 1.8**. The ‘0.List\_KW.csv’ file contains all the keywords which need to be collected, and the format can be seen as **Figure 1.8**. Put the ‘0.List\_KW.csv’ file and the code in the same folder.

For different collection time, we can create different folders to save the Twitter data. Each folder must contain both the ‘0.List\_KW.csv’ file and the data collection code. From the code, we can change the time periods (shown in **Figure 1.7 (b)**) to the expected date to save the Twitter data with different file name.

After all the files are prepared, run the code and the Twitter data will be collected in the same folder where code for data collection is kept seen as **Figure 1.8**.

A screenshot of a social media post

Description automatically generated

(a)

A screenshot of a social media post

Description automatically generated

(b)

**Figure 1.7** Example of Keyword Data Collection Code: (a) the code for input and output files; (b) the code for changing the time period of collection

A screen shot of a computer

Description automatically generated

**Figure 1.8** The example of 0.List\_KW.csv and output file

**2. Sentiment Analysis and Visualization**

**2.1 Sentiment Analysis**

The second part in the Jupyter Notebook is the sentiment analysis which can be seen under the heading ‘**3 Sentiment Analysis**’. The input of the sentiment analysis is the data collected from the part 1. From the codes, the ‘**path**’ defines the path of the input file folder and the ‘files’ is a list of the filename. Given the specific folder path, the output of this program is the sentiment analysis results. The sentiment analysis results contain the ‘user\_id’, ‘polarity’, and ‘subjectivity’. The examples can be seen as **Figure 2.1**.

After setting the input path, run the codes and the ‘**df\_final\_sentiment**’ is the output data frame. The example of input file path in the codes can be seen as **Figure 2.1 (a)** and the output file path can be seen as **Figure 2.1 (b)**.

A screenshot of a cell phone

Description automatically generated

(a)

A screenshot of a social media post

Description automatically generated

(b)

**Figure 2.1** Example of Sentiment Analysis Code: (a) the code for input files; (b) the code for output files

**2.2 Visualization of Sentiment Analysis Results**

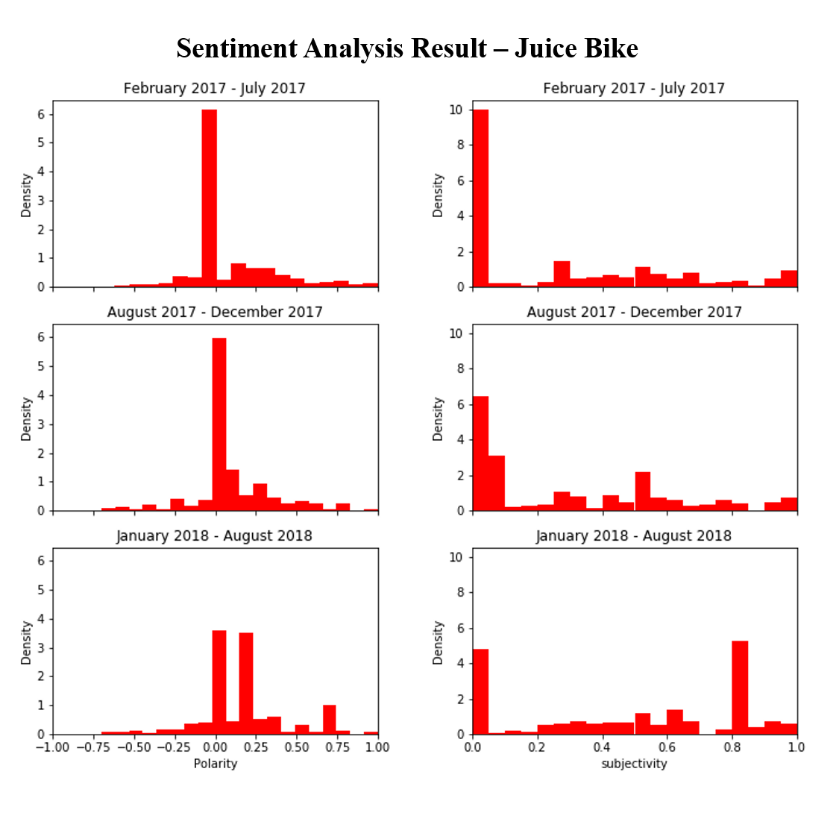
The visualization of sentiment analysis results can be seen as ‘**5 Sentiment Analysis Visualization**’ in the code file. The example of the code for visualizing the sentiment analysis is shown as **Figure 2.2**. From the code, we can change the path of input and output files as well as the time periods seen from **Figure 2.2**.

A screenshot of a cell phone

Description automatically generated

**Figure 2.2** Example of Sentiment Analysis Code: (a) the code for input files; (b) the code for output files

Run this code with the input file, the figure will be generated. **Figure 2.3** shows an example figure.

****

**Figure 2.3** Example of the visualization of sentiment analysis

**3. Topic Analysis and Visualization**

**3.1 Data Processing for Topic Analysis**

The third part in the Jupyter Notebook is the topic analysis. The first step of the topic analysis is to process the data which can be seen under the heading **‘7 Topic Model Data Processing’**, shown in Figure 3.1. From the code, we can change the path of input and output files to save the processed data for topic analysis. Some examples of input and output files are shown in **Figure 3.2**.

**A screenshot of a cell phone

Description automatically generated**

**Figure 3.1** Example of the topic analysis

A screenshot of a computer

Description automatically generated

**Figure 3.2** Example of the input and output files of topic analysis

**3.2 Prerequisite for Topic Analysis**

The code for topic analysis is written in Python 2 version which means that we need to use python 2 version to run the topic analysis. We use the Spyder IDE (in anaconda navigator) for python 2.7 version to apply the topic analysis. We can follow the instructions to install the required software and run the model.

* Download Anaconda (python 2.7) 32-bit Graphical Installer
* Open Spyder from the anaconda navigator (python 2.7) version
* Open the ldaModel.py
* Install all the necessary python packages
* Change the input path and file name
* Run the model

**3.3 Topic Analysis**

For the topic analysis, we use a tool based on Python 2.7 version environment. Thus, Python 2.7 is required in the topic analysis program. The procedure of topic analysis can be seen as follows:

* Download the ‘topic\_analysis\_src.rar’ archive and unzip the archive.
* Find the ldaModel.py file, which contains the code for topic analysis.
* Based on the ldaModel.py, install all the required python packages, seen as **Figure 3.3 (a)**.
* Process the raw Twitter data into the input file of topic analysis. The input file of topic analysis contains two column – ‘user\_id’ and ‘tweets’, which can be found in Figure 6.
* At the end of the codes (ldaModel.py), the ‘**data\_folder**’ (seen as **Figure 3.3 (b))** should be changed into the path where the ‘topic\_analysis\_src/model’ is and the ‘raw\_input\_file’ is the path of the input file. The example of input file can be seen as **Figure 3.4.**
* In the **runLDAmodel** function, ***k*** represents the number of topics. Set the number of topics by changing the value of ‘**k**’.
* After running all the above process, run the ldaModel.py and the results will be saved in the same path of the ‘topic\_analysis\_src’. The output file name is ‘RT\_LDA\_patterns’.

A screenshot of a social media post

Description automatically generated

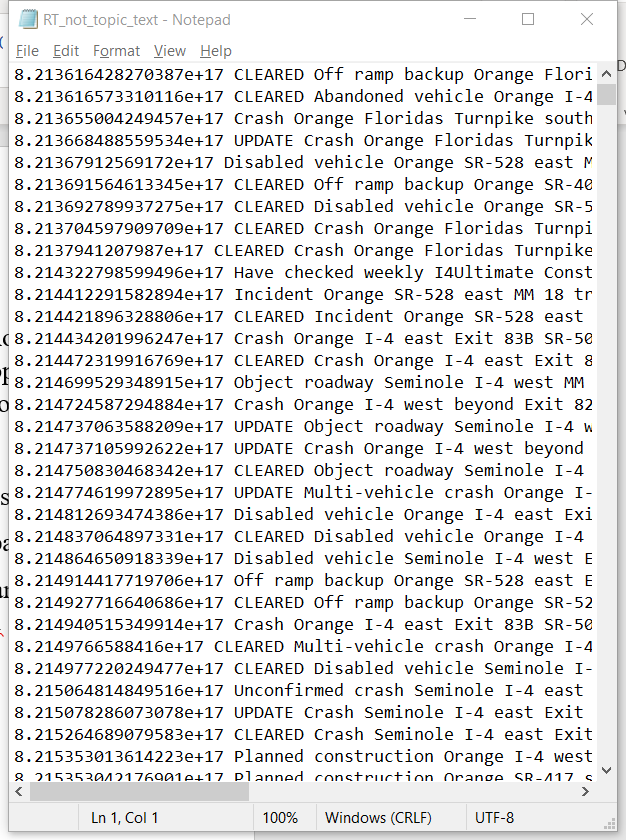
(a)

A screenshot of a cell phone

Description automatically generated

(b)

**Figure 3.3** Example of the topic analysis: (a) necessary packages; (b) input path



**Figure 3.4** Example of the input file

There are also three required files – dictionary.dat, sequence\_sanitized.dat and user.dat, seen as **Figure 3.5.**

A screenshot of a cell phone

Description automatically generated

**Figure 3.5** Example of required files and output file for topic analysis

**3.4 Topic Analysis Results Visualization**

The first step for visualization of the topic analysis results is to process the data which can be seen under the heading ‘**8 Data Processing for Visualization**’. In the codes, the ‘**path\_input**’ is the path of the input file (format of the input file can be seen as **Figure 3.7**). The code for topic analysis visualization can be seen as **Figure 3.8.** In the code, we should change the input file path to read the input file and output figure path to save the output figures.

A screenshot of a social media post

Description automatically generated

**Figure 3.6** Data Processing for Visualization of Topic Analysis

**A screenshot of a computer

Description automatically generated**

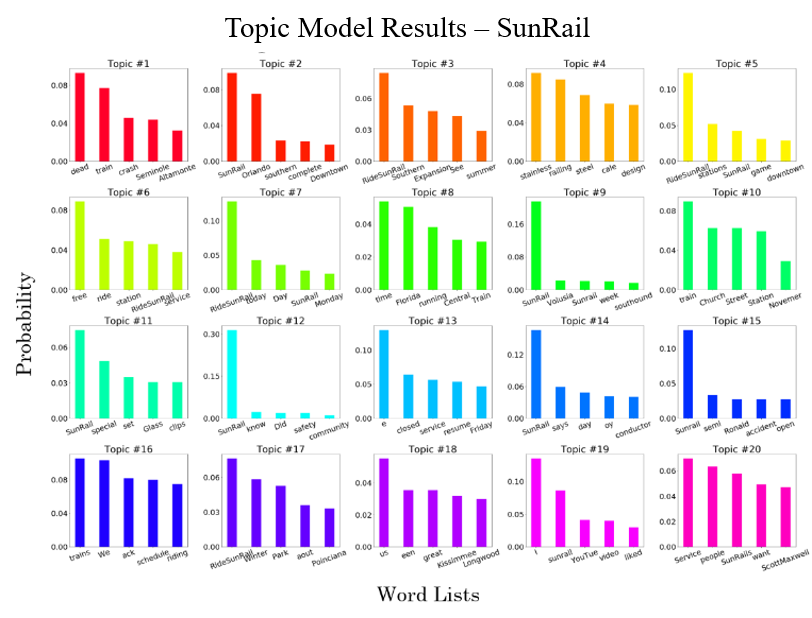
**Figure 3.7** Data Samples for Visualization of Topic Analysis

A screenshot of a social media post

Description automatically generated

**Figure 3.8** Example of Visualization of Topic Analysis

Run this code with the input file, the figure can be shown. **Figure 3.9** shows one of the examples.



**Figure 3.9** Example of the visualization of topic analysis