# SENTIMENT ANALYSIS FOR MARKETING

# BATCH MEMBER

**211121104054 : THULASI NAVANEETHAN . N**

**Phase 3 Submission Document**

**Phase 3 : Development Part 1**

**Topic : Start building the sentiment analysis model by loading and preprocessing the dataset .**

# Introduction To Sentiment Analysis

Sentiment analysis refers to analyzing an opinion or feelings about something using data like text or images, regarding almost anything. Sentiment analysis helps companies in their decision-making process. For instance, if public sentiment towards a product is not so good, a company may try to modify the product or stop the production altogether in order to avoid any losses.

There are many sources of public sentiment e.g. public interviews, opinion polls, surveys, etc. However, with more and more people joining social media platforms, websites like Facebook and Twitter can be parsed for public sentiment.

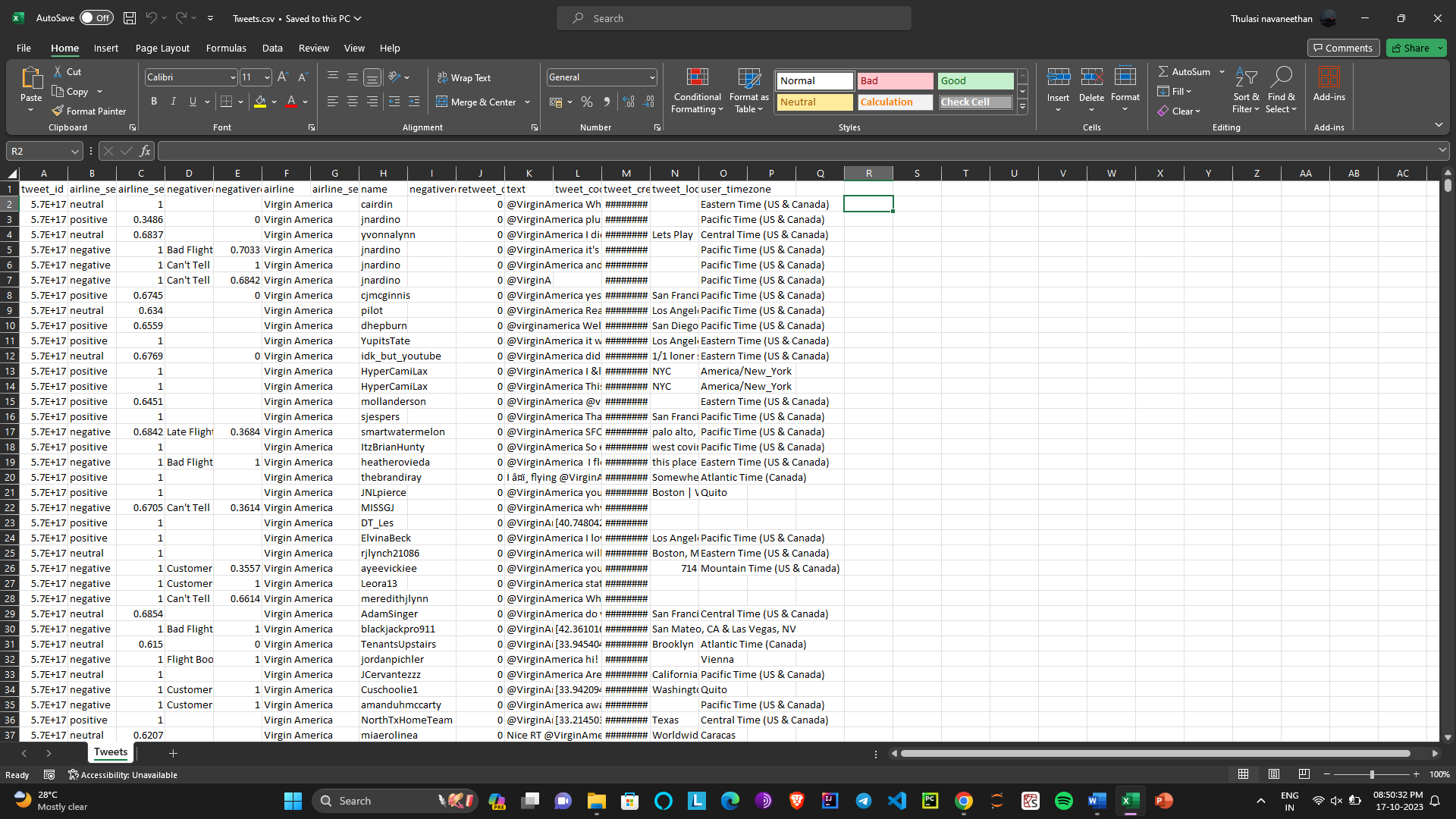
# Problem Definition

Given tweets about six US airlines, the task is to predict whether a tweet contains positive, negative, or neutral sentiment about the airline. This is a typical supervised learning task where given a text string, we have to categorize the text string into predefined categories.

# Solution

To solve this problem, we will follow the typical machine learning pipeline. We will first import the required libraries and the dataset. We will then do exploratory data analysis to see if we can find any trends in the dataset. Next, we will perform text preprocessing to convert textual data to numeric data that can be used by a machine learning algorithm. Finally, we will use machine learning algorithms to train and test our sentiment analysis models.

**Given dataset:**

****

**Importance Of loading and processing dataset :**

Loading and processing a dataset is a crucial step in data analysis, machine learning, and many other data-related tasks. The importance of this step lies in its role in ensuring the quality, integrity, and suitability of the data for subsequent analysis and modeling. Here are some key reasons why loading and processing a dataset is important:

1. Data Quality Assurance: Loading and processing the dataset allows you to identify and address data quality issues, such as missing values, outliers, inconsistencies, and errors. This is critical for ensuring the accuracy and reliability of any analysis or modeling that follows.

2. Data Cleaning : Often, real-world datasets are messy and contain missing or irrelevant information. Pre-processing involves cleaning the data by removing or imputing missing values and eliminating redundant or noisy features, making the dataset more suitable for analysis.

3. Data Transformation : You may need to transform data into a more appropriate format or scale. For example, normalizing features can be important for machine learning algorithms that are sensitive to the scale of input features.

4. Feature Engineering : Feature engineering involves creating new features or modifying existing ones to better represent the underlying patterns in the data. Proper feature engineering can significantly improve the performance of machine learning models.

5. Data Exploration : Loading the dataset allows you to explore its characteristics, distributions, and relationships between variables. This exploratory data analysis (EDA) helps you gain insights and inform subsequent analysis decisions.

6. Data Preprocessing for Machine Learning : In the context of machine learning, loading and preprocessing data are essential steps. You need to split the data into training and testing sets, perform one-hot encoding or label encoding for categorical variables, and handle class imbalances, if any.

7. Data Security and Privacy : Ensuring that sensitive information is properly handled and protected is crucial. Loading and processing data offer opportunities to anonymize or mask personally identifiable information to comply with data privacy regulations.

8. Data Reduction : Large datasets can be computationally expensive to work with. Preprocessing can involve dimensionality reduction techniques like PCA to reduce the number of features while preserving important information.

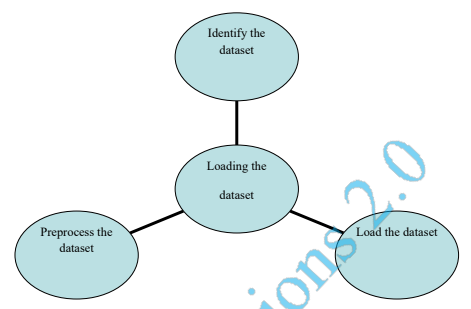
9. Data Standardization : In some cases, it's important to standardize data so that it can be easily compared or combined with other datasets. This involves ensuring consistent units, time zones, or data formats.

10. Data Visualization : Data visualization often comes after data loading and processing. It helps you communicate your findings and insights effectively and aids in understanding the data's characteristics.

11. Model Interpretability : Proper preprocessing can make your models more interpretable by ensuring that input features are meaningful and appropriately scaled.

12. Time and Resource Efficiency : Preprocessing can also involve optimizing data storage formats or compression techniques to save storage space and reduce the time required for data loading and analysis.

In summary, loading and processing a dataset are essential steps in any data-driven project. Properly prepared data ensures that subsequent analysis, modeling, and decision-making processes are accurate, efficient, and reliable. It also helps in uncovering valuable insights from the data and addressing issues of data quality, privacy, and compliance.



**Program :**

import numpy as np  
import pandas as pd  
import re  
import seaborn as sns  
import nltk  
import matplotlib.pyplot as plt  
%matplotlib inline

airline\_tweets = pd.read\_csv(r'/content/Tweets.csv')  
airline\_tweets.head()

tweet\_id airline\_sentiment airline\_sentiment\_confidence \  
0 570306133677760513 neutral 1.0000   
1 570301130888122368 positive 0.3486   
2 570301083672813571 neutral 0.6837   
3 570301031407624196 negative 1.0000   
4 570300817074462722 negative 1.0000   
  
 negativereason negativereason\_confidence airline \  
0 NaN NaN Virgin America   
1 NaN 0.0000 Virgin America   
2 NaN NaN Virgin America   
3 Bad Flight 0.7033 Virgin America   
4 Can't Tell 1.0000 Virgin America   
  
 airline\_sentiment\_gold name negativereason\_gold retweet\_count \  
0 NaN cairdin NaN 0   
1 NaN jnardino NaN 0   
2 NaN yvonnalynn NaN 0   
3 NaN jnardino NaN 0   
4 NaN jnardino NaN 0   
  
 text tweet\_coord \  
0 @VirginAmerica What @dhepburn said. NaN   
1 @VirginAmerica plus you've added commercials t... NaN   
2 @VirginAmerica I didn't today... Must mean I n... NaN   
3 @VirginAmerica it's really aggressive to blast... NaN   
4 @VirginAmerica and it's a really big bad thing... NaN   
  
 tweet\_created tweet\_location user\_timezone   
0 2015-02-24 11:35:52 -0800 NaN Eastern Time (US & Canada)   
1 2015-02-24 11:15:59 -0800 NaN Pacific Time (US & Canada)   
2 2015-02-24 11:15:48 -0800 Lets Play Central Time (US & Canada)   
3 2015-02-24 11:15:36 -0800 NaN Pacific Time (US & Canada)   
4 2015-02-24 11:14:45 -0800 NaN Pacific Time (US & Canada)

Let's explore the dataset a bit to see if we can find any trends. But before that, we will change the default plot size to have a better view of the plots.

plot\_size = plt.rcParams["figure.figsize"]  
print(plot\_size[0])  
print(plot\_size[1])  
  
plot\_size[0] = 8  
plot\_size[1] = 6  
plt.rcParams["figure.figsize"] = plot\_size

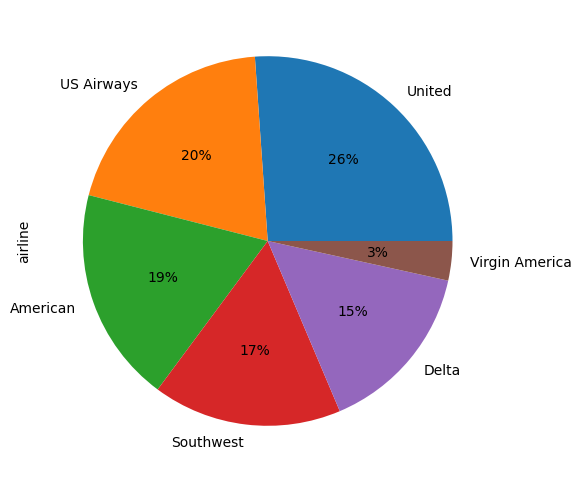
8.0  
6.0

# Exploration Of Data

## Let's first see the number of tweets for each airline. We will plot a pie chart for that:

airline\_tweets.airline.value\_counts().plot(kind='pie', autopct='%1.0f%%')

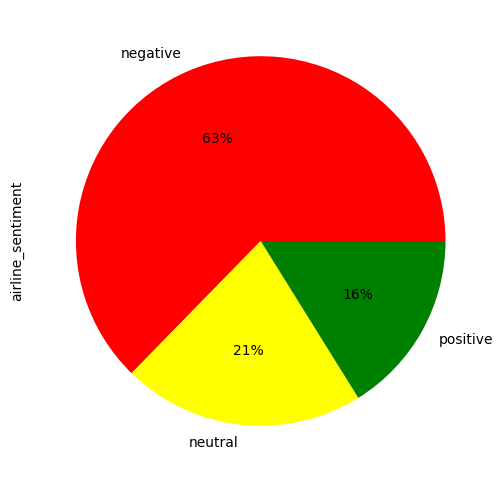
<Axes: ylabel='airline'>



Let's now see the distribution of sentiments across all the tweets.

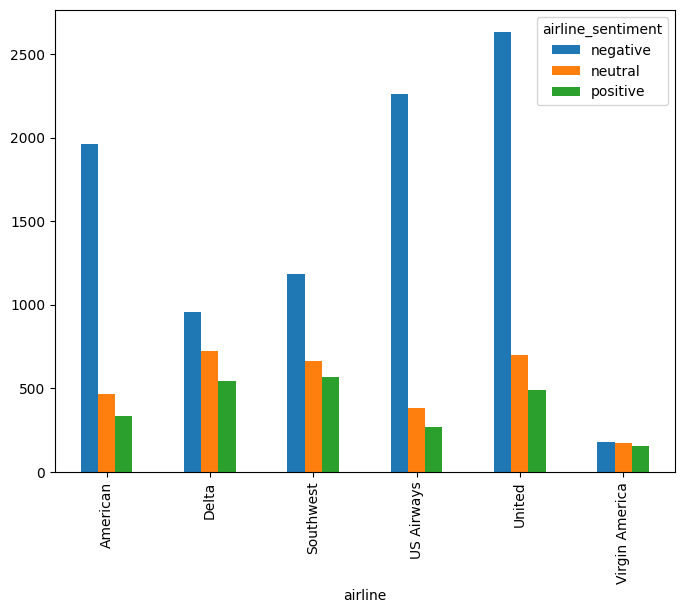
airline\_tweets.airline\_sentiment.value\_counts().plot(kind='pie', autopct='%1.0f%%', colors=["red", "yellow", "green"])

<Axes: ylabel='airline\_sentiment'>



airline\_sentiment = airline\_tweets.groupby(['airline', 'airline\_sentiment']).airline\_sentiment.count().unstack()  
airline\_sentiment.plot(kind='bar')

<Axes: xlabel='airline'>

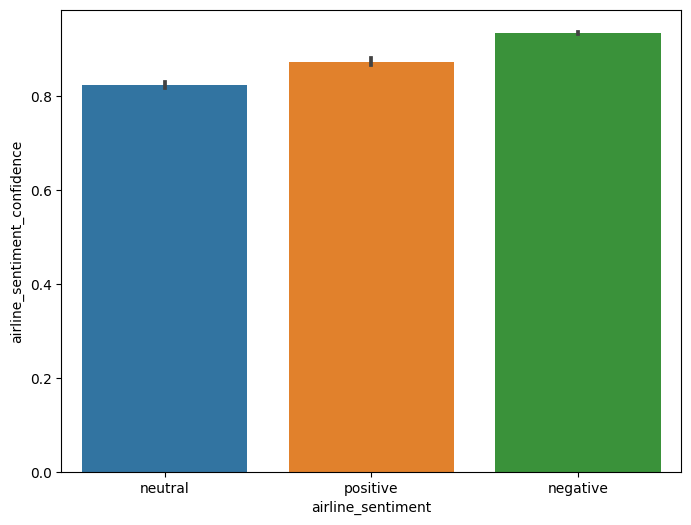


It is evident from the output that for almost all the airlines, the majority of the tweets are negative, followed by neutral and positive tweets. Virgin America is probably the only airline where the ratio of the three sentiments is somewhat similar.

Finally, let's use the Seaborn library to view the average confidence level for the tweets belonging to three sentiment categories.

sns.barplot(x='airline\_sentiment', y='airline\_sentiment\_confidence' , data=airline\_tweets)

<Axes: xlabel='airline\_sentiment', ylabel='airline\_sentiment\_confidence'>



From the output, you can see that the confidence level for negative tweets is higher compared to positive and neutral tweets.

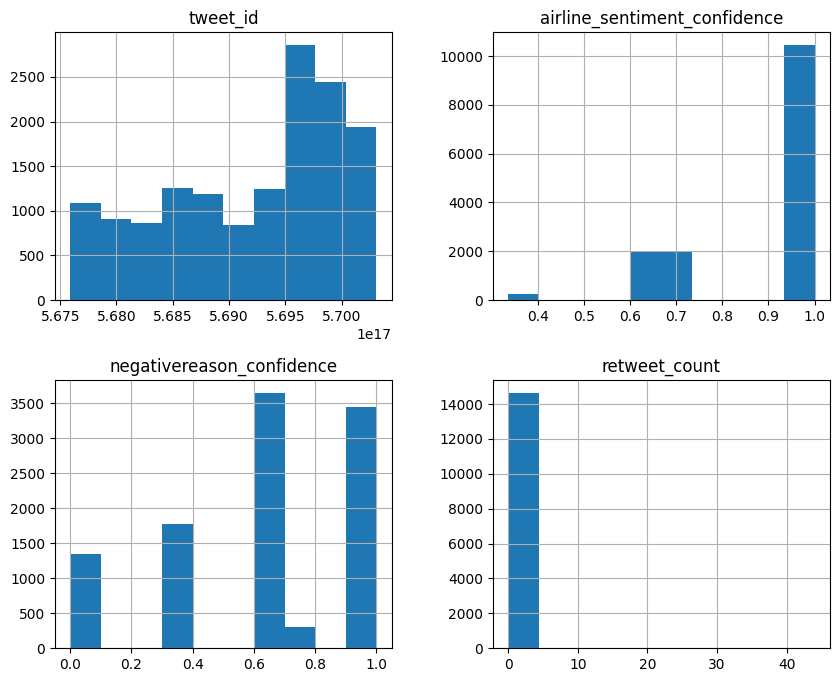
# Data Cleaning

Tweets contain many slang words and punctuation marks. We need to clean our tweets before they can be used for training the machine learning model. However, before cleaning the tweets, let's divide our dataset into feature and label sets.

Our feature set will consist of tweets only. If we look at our dataset, the 11th column contains the tweet text. Note that the index of the column will be 10 since pandas columns follow zero-based indexing scheme where the first column is called 0th column. Our label set will consist of the sentiment of the tweet that we have to predict. The sentiment of the tweet is in the second column (index 1). To create a feature and a label set, we can use the iloc method off the pandas data frame.

airline\_tweets.hist(figsize=(10,8))

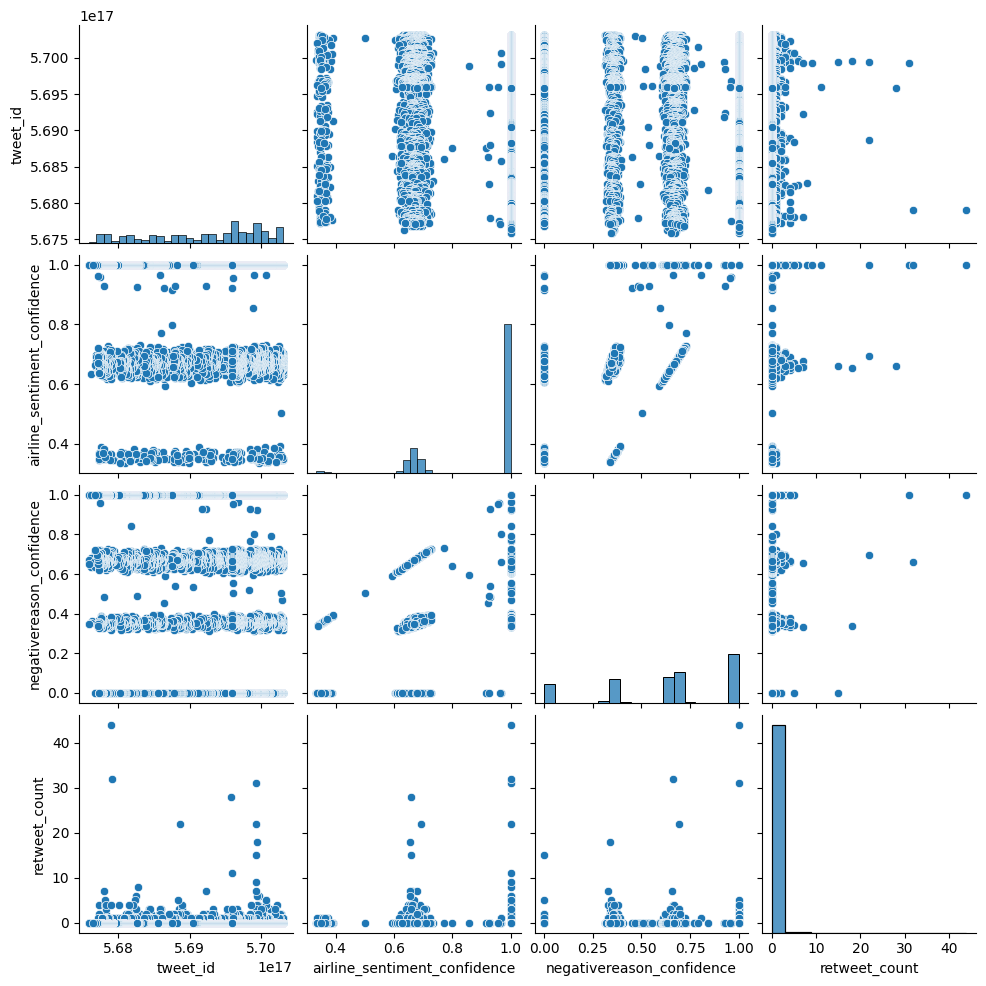
array([[<Axes: title={'center': 'tweet\_id'}>,  
 <Axes: title={'center': 'airline\_sentiment\_confidence'}>],  
 [<Axes: title={'center': 'negativereason\_confidence'}>,  
 <Axes: title={'center': 'retweet\_count'}>]], dtype=object)



plt.figure(figsize=(12,8))  
sns.pairplot(airline\_tweets)

<seaborn.axisgrid.PairGrid at 0x787bf77a6a10>

<Figure size 1200x800 with 0 Axes>



airline\_tweets.corr(numeric\_only=True)

tweet\_id airline\_sentiment\_confidence \  
tweet\_id 1.000000 0.024840   
airline\_sentiment\_confidence 0.024840 1.000000   
negativereason\_confidence 0.021533 0.685879   
retweet\_count -0.008852 0.012581   
  
 negativereason\_confidence retweet\_count   
tweet\_id 0.021533 -0.008852   
airline\_sentiment\_confidence 0.685879 0.012581   
negativereason\_confidence 1.000000 0.021574   
retweet\_count 0.021574 1.000000

plt.figure(figsize=(10,5))  
sns.heatmap(airline\_tweets.corr(numeric\_only = True), annot=True)

<Axes: >



**Conclusion** :

In conclusion, sentiment analysis models are powerful tools for extracting valuable insights from text data, enabling businesses, researchers, and individuals to understand and harness sentiment in a wide range of applications.