Sentiment analysis for marketing

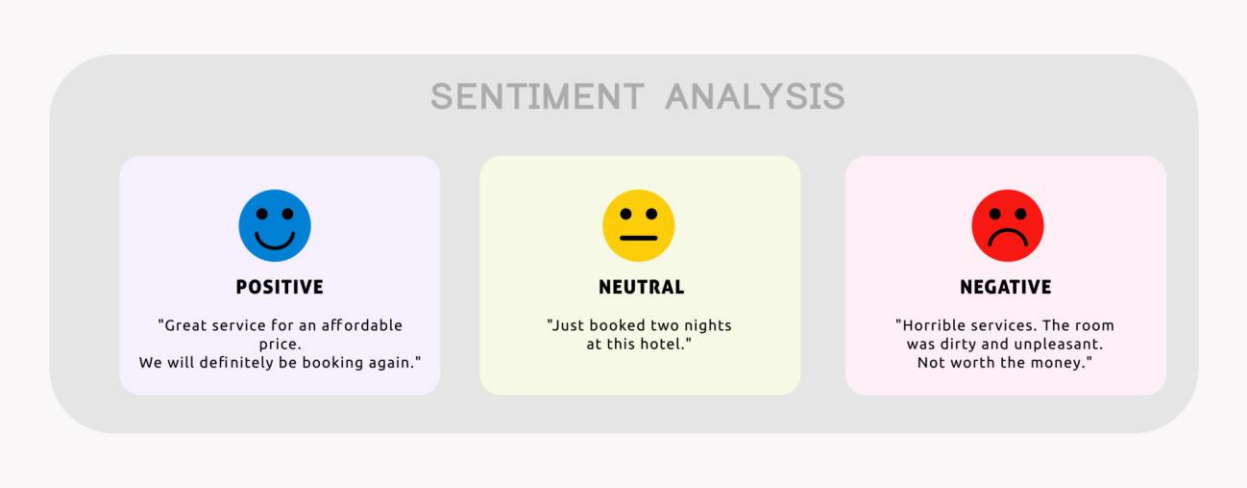
PUTTU PAVAN KUMAR

Phase-3 Document Submission

Project : Sentiment analysis for marketing

Phase 3: Development Part 1

Topic : In this part you will begin building your project by loading and preprocessing the dataset. Start building the sentiment analysis solution by loading dataset and preprocessing the data.



Introduction :

Sentiment analysis is a powerful tool that can be used to understand customer sentiment and make better marketing decisions. It uses natural language processing (NLP) and machine learning to extract opinions and

emotions from text data. This data can come from a variety of sources, such as social media posts, customer reviews, and survey responses.

Sentiment analysis can be used to answer a variety of marketing questions, such as:

- What do customers think of our products and services?
- What are the strengths and weaknesses of our brand?
- What are customers saying about our competitors?
- How effective are our marketing campaigns?
- What can we do to improve our customer satisfaction?

Sentiment analysis can be used to improve marketing campaigns in a number of ways. For example, it can be used to:

- Identify target audiences and develop more relevant marketing messages.
- Monitor the performance of marketing campaigns and make adjustments as needed.
- Respond to customer feedback in a timely and effective manner.
- Identify and address customer pain points.
- Improve product and service development.

Overall, sentiment analysis is a valuable tool for marketers who want to better understand their customers and improve their marketing efforts.

Here are some specific examples of how sentiment analysis can be used for marketing projects:

- A company can use sentiment analysis to analyze customer reviews on social media to identify common complaints and areas for improvement.
- A business can use sentiment analysis to track the performance of a new marketing campaign and see how customers are responding to it.

- A brand can use sentiment analysis to compare itself to its competitors and see how it fares in terms of customer satisfaction.
- A product manager can use sentiment analysis to gather feedback from customers about a new product design.
- A marketing team can use sentiment analysis to segment its email list and send more targeted messages to different groups of customers.

Sentiment analysis is a powerful tool that can be used to improve marketing campaigns in a variety of ways. By understanding what customers are saying and feeling about a brand, businesses can make better decisions about how to reach and engage with them.

Given dataset :

tweet_id	airline_sen...	airline_sen...	negativere...	negativere...	airline
570306133677760513	neutral	1.0			Virgin Amer
570301130888122368	positive	0.3486		0.0	Virgin Amer
570301083672813571	neutral	0.6837			Virgin Amer
570301031407624196	negative	1.0	Bad Flight	0.7033	Virgin Amer
570300817074462722	negative	1.0	Can't Tell	1.0	Virgin Amer
570300767074181121	negative	1.0	Can't Tell	0.6842	Virgin Amer
570300616901320704	positive	0.6745		0.0	Virgin Amer
570300248553349120	neutral	0.634			Virgin Amer
570299953286942721	positive	0.6559			Virgin Amer
570295459631263746	positive	1.0			Virgin Amer
570294189143031808	neutral	0.6769		0.0	Virgin Amer

Importance of loading and processing dataset :

Loading and processing datasets are fundamental steps in data analysis, machine learning, and many other data-driven applications. Here's why these steps are so important:

Data Accessibility:

Before any analysis can be performed, data must be loaded into an environment where it can be manipulated. This often means reading from databases, files, or external sources.

Data Quality:

Real-world data can be messy. It might contain missing values, duplicates, and outliers.By processing the data, you can clean and transform it, ensuring its quality and reliability for subsequent analysis or modeling.

Feature Engineering:

Once data is loaded, often you'll need to create new features from the existing ones to better capture the underlying patterns in the data.For example, from a timestamp, you might extract the hour of the day, day of the week, or even whether it's a holiday.

Data Scaling and Normalization:

Many machine learning algorithms work better when numerical features have the same scale. By processing the data, you can apply scaling or normalization to ensure that all features have values in a similar range.

Data Integration:

Often, data comes from multiple sources. Loading and processing allow you to integrate these diverse datasets, ensuring consistent and unified information.

Ensuring Data Privacy:

When processing data, especially personally identifiable information (PII), you may need to anonymize or encrypt certain fields to ensure privacy and compliance with regulations.

Efficiency and Performance:

Large datasets can be unwieldy and slow down analysis or training. By loading data efficiently (e.g., using appropriate data structures) and processing it (e.g., by filtering irrelevant records), you can ensure more timely and responsive operations.

Challenges involved in loading and preprocessing sentiment analysis dataset :

Loading and preprocessing sentiment analysis datasets involve some unique challenges. While many of the general challenges associated with data loading and preprocessing apply here as well, sentiment analysis has its peculiarities:

Language and Slang: Natural languages evolve, and new slang words emerge. A sentiment analysis model trained on older data may struggle to understand newer phrases or words.

Handling Ambiguity: Natural language is often ambiguous. For instance, sarcasm is a form of speech where the intended sentiment might be opposite to the literal words.

Imbalanced Datasets: Often, sentiment analysis datasets are imbalanced, with more instances of one sentiment class than others. This imbalance can lead to biased model training if not addressed.

Multilingual Data: If the dataset contains multiple languages, preprocessing steps like tokenization or stemming might require language-specific tools.

Data Annotation Consistency: Manual annotations can be inconsistent due to the subjective nature of sentiment. Two annotators might label the same text differently based on their interpretation.

Short Texts: Tweets or other short text snippets may not provide a lot of contextual information, making preprocessing and analysis more challenging.

Noise in the Data: User-generated content, which is often the source for sentiment analysis datasets, contains noise like URLs, usernames, emojis, and misspellings.

Context Dependency: The sentiment of a statement can be context-dependent. For instance, "This is sick!" can be positive in one context and negative in another.

Data Privacy: User-generated content might contain personally identifiable information, and this poses privacy concerns. Such data must be anonymized or removed.

Tokenization Challenges: Especially in languages where whitespace does not separate words, tokenization can be complex.

1.Loading the dataset:

Loading the dataset using machine learning is the process of bringing

the data into the machine learning environment so that it can be used

to train and evaluate a model.

The specific steps involved in loading the dataset will vary depending

on the machine learning library or framework that is being used.

However, there are some general steps that are common to most

machine learning frameworks:

a.Identify the dataset:

The first step is to identify the dataset that you want to load. This dataset may be stored in a local file, in a database, or in a cloud storage service.

b.Load the dataset:

Once you have identified the dataset, you need to load it into the machine learning environment. This may involve using a built-in function in the machine learning library, or it may involve writing your own code.

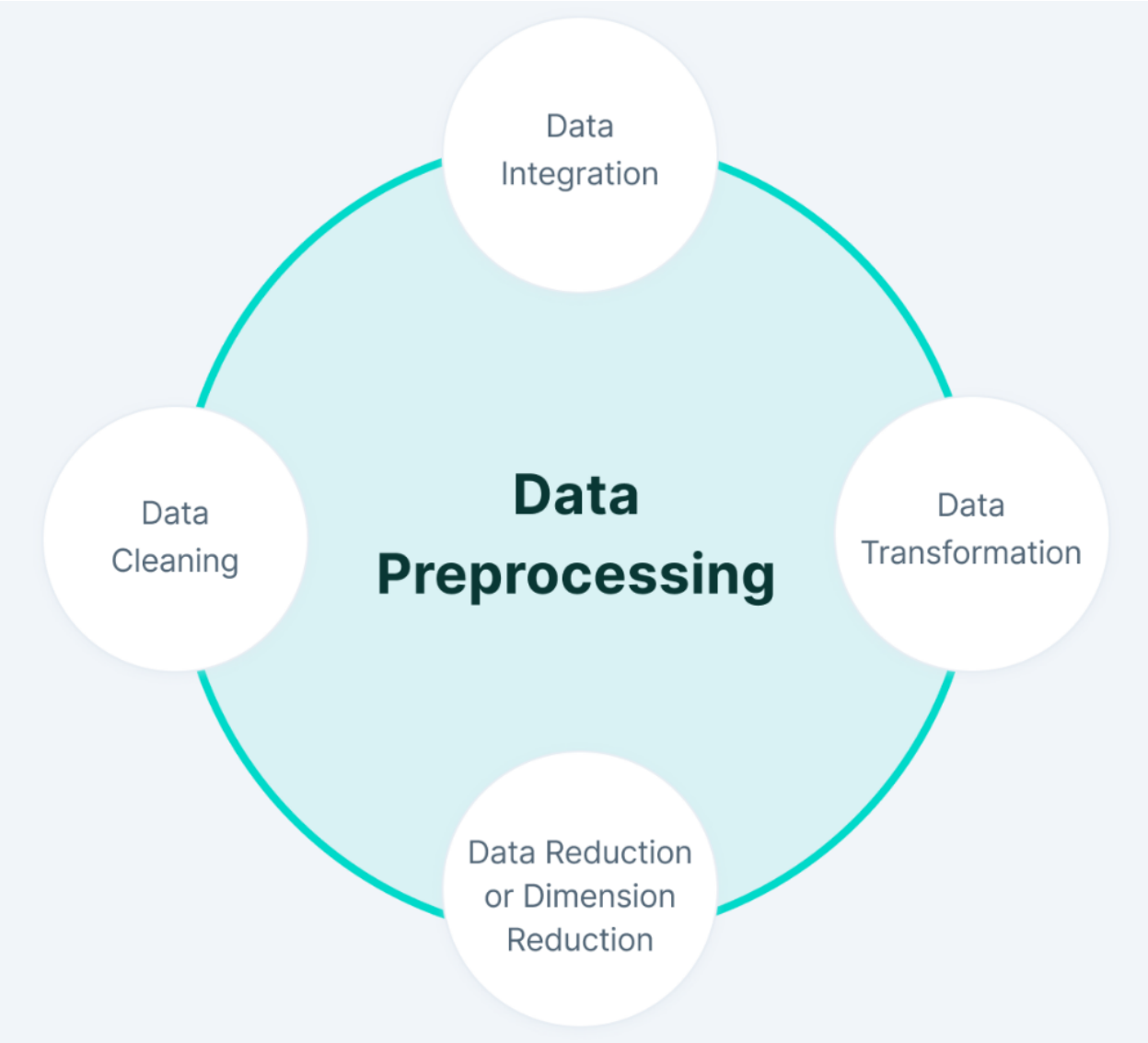
c.Preprocess the dataset:

Once the dataset is loaded into the machine learning environment, you may need to preprocess it before you can start training and evaluating your model. This may involve cleaning the data, transforming the data into a suitable format and splitting the data into training and test sets.

2.Data Preprocessing

Data preprocessing is the critical first step in any machine learning project. It involves cleaning the data, removing outliers, and handling missing

values to prepare the dataset for model training. In the context of the house price prediction project, let's elaborate on the specific steps:



Python Program :

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```
import seaborn as sns
import missingno as msno
import warnings
warnings.filterwarnings(action='ignore')

# Import NLTK and download required resources
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize, sent_tokenize
from nltk.stem import LancasterStemmer, WordNetLemmatizer

nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')

# Import other libraries
import re
import string
import unicodedata
import contractions
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
import wordcloud
from wordcloud import STOPWORDS, WordCloud
import pandas as pd
from sklearn.model_selection import train_test_split, StratifiedKFold
from sklearn.svm import LinearSVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
from sklearn.metrics import (
    recall_score,
    accuracy_score,
    confusion_matrix,
    classification_report,
    f1_score,
    precision_score,
    precision_recall_fscore_support
)

# Set options for displaying data
pd.set_option("display.max_columns", None)
pd.set_option("display.max_rows", 200)
```

Output :

tweet_id	airline_sen...	airline_sen...	negativere...	negativere...	airline
570306133677760513	neutral	1.0			Virgin Amer
570301130888122368	positive	0.3486		0.0	Virgin Amer
570301083672813571	neutral	0.6837			Virgin Amer
570301031407624196	negative	1.0	Bad Flight	0.7033	Virgin Amer
570300817074462722	negative	1.0	Can't Tell	1.0	Virgin Amer
570300767074181121	negative	1.0	Can't Tell	0.6842	Virgin Amer
570300616901320704	positive	0.6745		0.0	Virgin Amer
570300248553349120	neutral	0.634			Virgin Amer
570299953286942721	positive	0.6559			Virgin Amer
570295459631263746	positive	1.0			Virgin Amer
570294189143031808	neutral	0.6769		0.0	Virgin Amer

Observation :

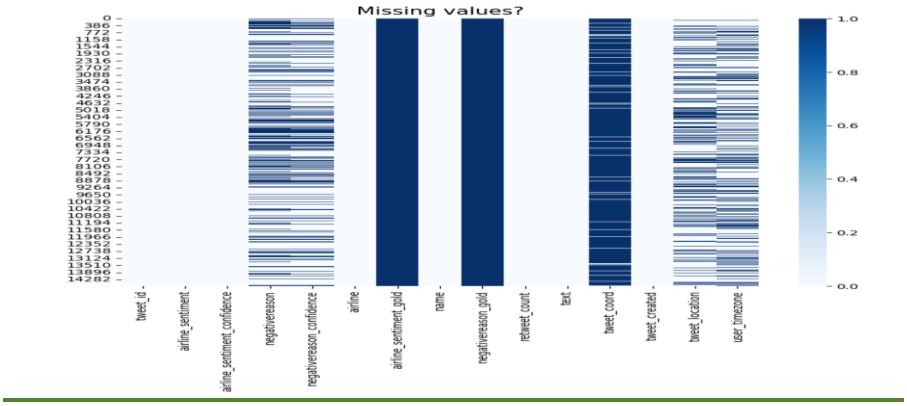
There are 15 columns in the dataset. Half of the columns have null values. Considering both dependent and independent variables not having any null values, we will not do any null value processing. Most columns in the dataset are of object type. airline_sentiment is our dependent / target variable. text column is our independent variable that we will use for analysis. All other columns will be dropped at a later stage.

```
df.isnull().sum()
```

```
tweet_id                          0
airline_sentiment                 0
airline_sentiment_confidence      0
negativereason                 5462
```

```
negativereason_confidence        4118
airline                             0
airline_sentiment_gold          14600
name                                0
negativereason_gold             14608
retweet_count                       0
text                                0
tweet_coord                     13621
tweet_created                       0
tweet_location                  4733
user_timezone                   4820
dtype: int64
```

```
#Visualization of missing value using heatmap
plt.figure(figsize=(10,7))
sns.heatmap(df.isnull(), cmap = "Blues")
plt.title("Missing values?", fontsize = 15)
plt.show()
```




```
print("Percentage null or na values in df")
((df.isnull() | df.isna()).sum() * 100 / df.index.size).round(2)
```

```
Percentage null or na values in df
```

```
tweet_id                        0.00
airline_sentiment               0.00
airline_sentiment_confidence    0.00
negativereason                 37.31
negativereason_confidence      28.13
airline                         0.00
airline_sentiment_gold         99.73
name                            0.00
negativereason_gold            99.78
retweet_count                   0.00
text                            0.00
```

```
tweet_coord                   93.04
tweet_created                  0.00
tweet_location                32.33
user_timezone                 32.92
dtype: float64
```
linkcode

```
df.drop(["tweet_coord", "airline_sentiment_gold", "negativereason_gold"], axis=1, inp
lace=True)
```

```
linkcode
df.head()
```

	tweet_id	airline_sentiment	airline_sentiment_confidence	negativereason	negativereason_confidence	airline
0	570306133677760513	neutral	1.0000	NaN	NaN	Virgin America
1	570301130888122368	positive	0.3486	NaN	0.0000	Virgin America
2	570301083672813571	neutral	0.6837	NaN	NaN	Virgin America
3	570301031407624196	negative	1.0000	Bad Flight	0.7033	Virgin America
4	570300817074462722	negative	1.0000	Can't Tell	1.0000	Virgin America

```
freq = df.groupby("negativereason").size()
```

```
df.duplicated().sum()
```
```
# Dropping duplicates
df.drop_duplicates(inplace = True)
```

```
df.duplicated().sum()
```
```
linkcode
df.describe().T
```

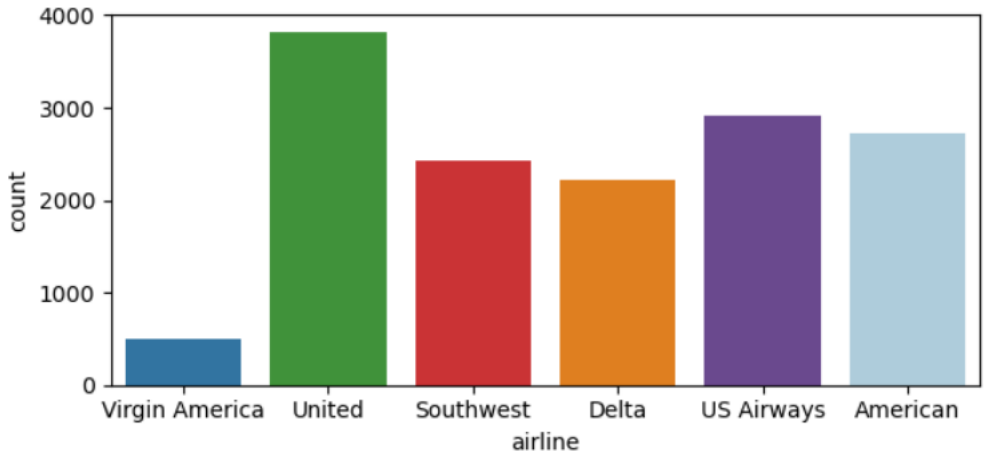
	count	mean	std	min	25%	50%
tweet_id	14601.0	5.692156e+17	7.782706e+14	5.675883e+17	5.685581e+17	5.694720e+17
airline_sentiment_confidence	14601.0	8.999022e-01	1.629654e-01	3.350000e-01	6.923000e-01	1.000000e+00
negativereason_confidence	10501.0	6.375749e-01	3.303735e-01	0.000000e+00	3.605000e-01	6.705000e-01
retweet_count	14601.0	8.280255e-02	7.467231e-01	0.000000e+00	0.000000e+00	0.000000e+00

EDA :

```
df.nunique()
```

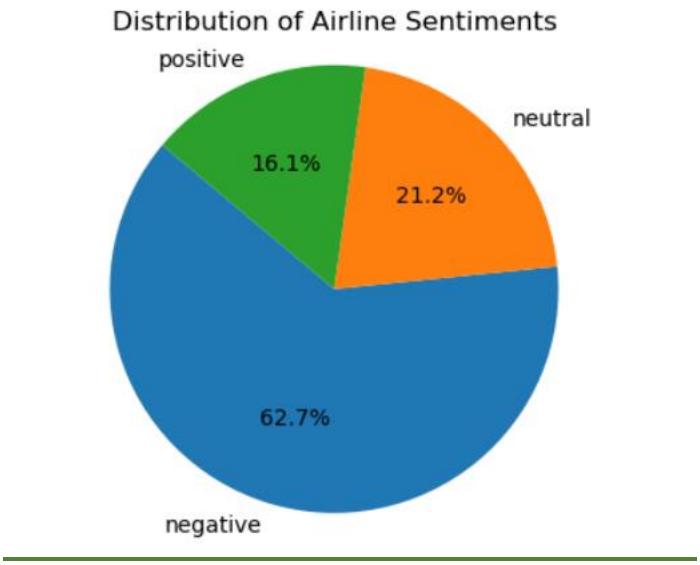
```
tweet_id                      14485
airline_sentiment                 3
airline_sentiment_confidence   1023
negativereason                   10
negativereason_confidence      1410
airline                           6
name                           7701
retweet_count                    18
text                          14427
tweet_created                 14247
tweet_location                 3081
user_timezone                    85
dtype: int64
```

```
# Checking the distribution of airlines
plt.figure(figsize=(7,3))
sns.countplot(data=df,x='airline', palette=['#1f78b4', '#33a02c', '#e31a1c', '#ff7f00
', '#6a3d9a', '#a6cee3'])
plt.show()
```



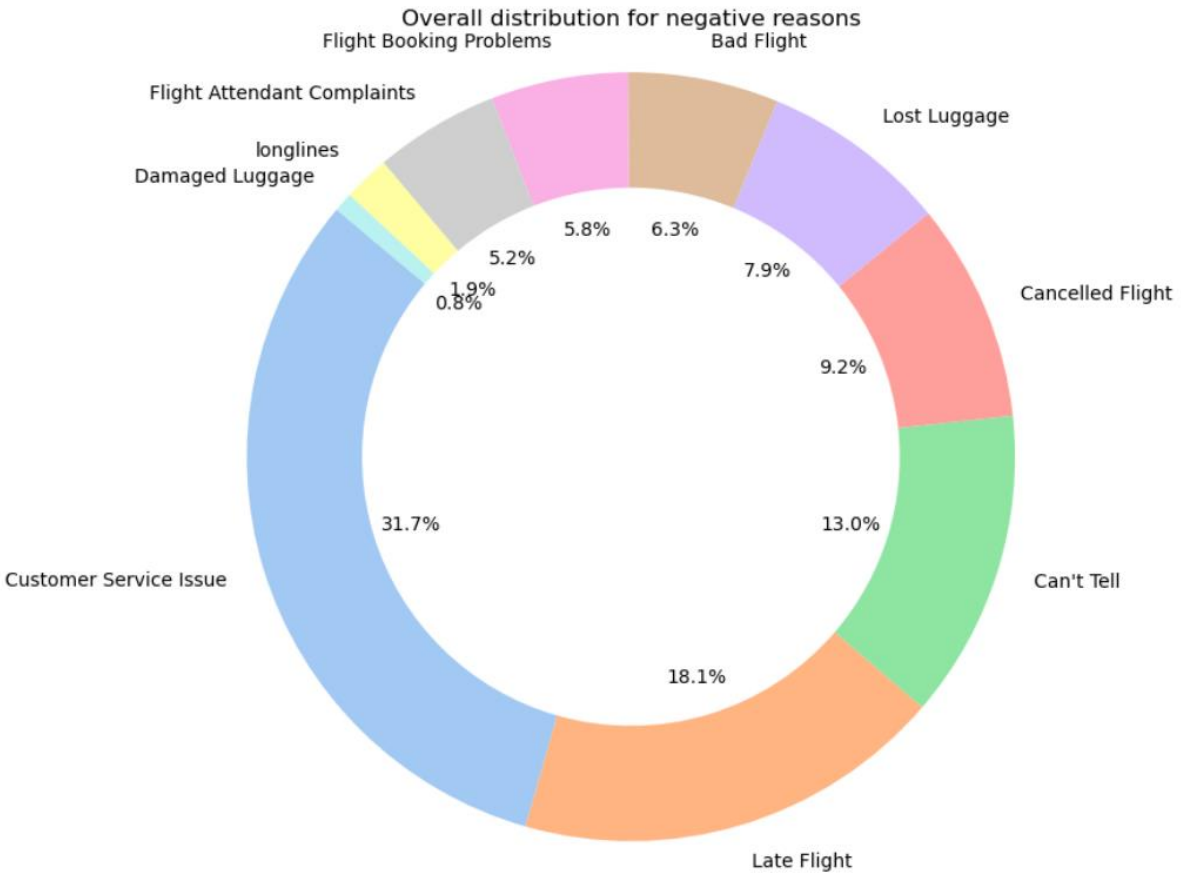

```
# Visualize the distribution of airline sentiments using a pie chart
sentiment_counts = df['airline_sentiment'].value_counts()
plt.figure(figsize=(6, 4))
```

```
plt.pie(sentiment_counts, labels=sentiment_counts.index, autopct='%1.1f%%', start
angle=140)
plt.title('Distribution of Airline Sentiments')
plt.axis('equal')  # Equal aspect ratio ensures that pie is drawn as a circle.
plt.show()
```




```
# Calculate the value counts for each negative reason
value_counts = df['negativereason'].value_counts()

# Create a donut-like pie chart using matplotlib and seaborn
plt.figure(figsize=(8, 8))
labels = value_counts.index
values = value_counts.values
colors = sns.color_palette('pastel')[0:len(labels)]  # Use pastel colors for the
chart
plt.pie(values, labels=labels, colors=colors, autopct='%1.1f%%', startangle=140,
wedgeprops=dict(width=0.3))
plt.title('Overall distribution for negative reasons')
plt.axis('equal')  # Equal aspect ratio ensures the pie chart is drawn as a circl
e.
plt.show()
```



Conclusion :

The sentiment analysis project can be a valuable tool for businesses to gain insights into customer sentiment towards competitor products. By understanding customer sentiments, businesses can identify strengths and weaknesses in competing products, thereby improving their own offerings.